Journal of Computers Vol. 32 No. 5, 2021, pp. 148-160
doi:10.53106/199115992021103205012

Research of Improved Whale Optimization Algorithm Based on Variable Convergence Factor and Forced Global Search

Cheng Zhu, Xu-Hua Pan*, Qi Chen, Yong Zhang, Xin-Yi Gao

Department of Information Engineering, Tianjin University of Commerce, Tianjin, P. R. China
{zhucheng, panxuhua, chenqi, zhangyong, gaoxinyi}@tjcu.edu.cn

Received 30 January 2021; Revised 12 May 2021; Accepted 16 May 2021

Abstract. Optimization Algorithm is new meta-heuristic optimization algorithm which mimicking the hunting behavior of humpback whales. It enjoys the advantages of simple principle, less parameters, remarkable search ability and good global convergence, also suffer many defects, such as slow convergence speed, low convergence precision and easy to fall into the local optimum. This paper analyzes the problems of original WOA, making use of Good-Point Set to generate initial population. Through variable convergence factor, the progress of search is more flexible and pertinent. At the same time, mechanism of forced global search makes the ability of jumping out local optimum is promoted in substance. Results and convergence curves of benchmark functions indicate that exploration, exploitation, local optima avoidance of algorithm in this paper are competitive with original WOA.

Keywords: whale optimization algorithm, good-point set, nonlinear variable convergence factor, meta-heuristic, swarm optimization

1 Introduction

Meta-heuristic algorithms are widely used in pattern recognition, system control, signal processing and other fields. Whale Optimization Algorithm (WOA) as new meta-heuristic optimization algorithm is proposed by Seyedali Mirjalili in 2016 [1]. It enjoys advantages of simple principle, less parameters, remarkable search ability and good global convergence. However, similar to other swarm intelligence algorithms, WOA also suffers many defects, such as slow convergence speed, low convergence precision and being easy to fall into the local optimum.

In order to settle above issues, there are many modified WOA proposed by scholars. Liu Zhusong's Whale Optimization Algorithm based on Chaotic Sine Cosine (CSCWOA), cosine local exploitation accelerates the speed of convergence, chaotic factor improves the ability of jumping out of local optimum [2]. Sun Yongjun delivers a modified whale optimization algorithm for large-scale global optimization problems [3]. Nonlinear dynamic strategy, lévy-flight and quadratic interpolation methods are applied to enhance the local exploitation ability and solution accuracy. Abdalla Mostafa proposes liver segmentation in MRI images based on whale optimization algorithm [4]. Xu Yufei presents a kind of WOA based on differential evolution and elite opposition-based learning (DEOBWOA) to optimize the kinetic model parameters of refinery [5]. Niu Peifeng's an integrated modeling method was proposed by combination of oppositely adaptive whale optimization algorithm (AWOA) with opposition-based learning and inertia weight factor [6]. It optimizes fast learning network.

This paper makes use of Good-Point Set, nonlinear variable convergence factor and mechanism named forced global search to promote the convergence speed and optimization accuracy of WOA. For convenience, the improved WOA researched in this paper is named as VCFWOA in later statements.

* Corresponding Author

2 Overview of Whale Optimization Algorithm (WOA)

The WOA is inspired from feeding habits of humpback whales. In this section, inspiration of the proposed method and mathematical model is described briefly.

Humpback whales' feeding and hunting behavior can be illustrated through 3 phases: Search for prey, encircling prey, spiral bubble-net attack.

2.1 Search for Prey (Exploration Phase)

WOA update the position of a search agent in the exploration phase according to a randomly chosen search agent. The mathematical model is as follows:

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A} \cdot \left| \vec{C} \cdot \overrightarrow{X_{rand}} - \vec{X} \right| \quad \textbf{(1)}$$

Where $\vec{X}$ is current position vector, and $\overrightarrow{X_{rand}}$ is a random position vector chosen from the population, t is the index for the current iteration. In Equation (1), $\vec{A}$ and $\vec{C}$ correspond to coefficient vectors which are defined as:

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a} \quad \textbf{(2)}$$

$$\vec{C} = 2 \cdot \vec{r} \quad \textbf{(3)}$$

Where vector $\vec{r}$ comprises uniformly distributed random numbers [0, 1], and $\vec{a}$ corresponds to a linearly reducing vector from 2 to 0 with the progression of iteration.

2.2 Encircling Prey

The location of the best search agent in a particular iteration is assumed to be the location of the prey. The other agents update positions towards the best search agent to encircle it. It can be modelled as:

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A} \cdot \left| \vec{C} \cdot \overrightarrow{X^*}(t) - \vec{X}(t) \right| \quad \textbf{(4)}$$

Where $\overrightarrow{X^*}$ denotes the location corresponding to the best agent. During each iteration, $\overrightarrow{X^*}$ is updated if any other whale finds a better location. Equation (4) allows any search agent to update its position in the neighborhood of the current best solution and simulates encircling the prey.

2.3 Bubble-net Attacking (Exploitation Phase)

Bubble-net attacking includes two approaches, shrinking encircling mechanism and spiral updating position. Where $\overrightarrow{D'} = |\overrightarrow{X^*}(t) - \vec{X}(t)|$ indicates the distance of the No. i search agent to best search agent obtained so far. p is a random number in [0, 1] to ensure there are fair chances between shrinking position updates and spiral. The mathematical model is as follows:

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A} \cdot \vec{D} & if\ p < 0.5 \\ \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) & if\ p \geq 0.5 \end{cases} \quad \textbf{(5)}$$

$|\vec{A}| < 1$ force search agent to move far away from a reference whale, in the contrast, $|\vec{A}| \geq 1$ emphasize exploration and allow the WOA algorithm to perform a global search.

3 Elaboration of VCFWOA

In order to overcome the defects as well as improve the performance of original WOA, this paper takes tasks from initial population, convergence state, convergence factor and forced global search.

3.1 Good-Point Set and Initial Population

In this section, the definition and mathematical model of Good-Point Set are first provided. The comparison of distribution of initial population generated by random number in original WOA and Good-Point Set is then proposed.

High quality initialization of population is very helpful to promote accuracy of algorithm and speed of convergence [7]. However, the original WOA algorithm gets initialization by way of random number, which is not capable of guaranteeing the diversity of population. This paper makes use of Good-Point Set to generate the initial population. The theory of Good-Point Set is proposed by Hua Luogeng who is a Chinese mathematician in 1978 [8]. The definition is as follow:

(1) Suppose G_s is the unit cube in s-dimension Euclidean space, $x \in G_s$, $x = (x_1, x_2, \ldots, x_s)$, $0 \le x_i \le 1$, $i = 1, 2, \ldots, s$.

(2) Suppose there is a points set (n points) in G_s .

$$P_n(k) = \{x_1^{(n)}(k), \ldots, x_s^{(n)}(k), 1 \le k \le n\}, 0 \le x_i^{(n)}(k) \le 1, 1 \le i \le s$$

(3) Suppose $r = (r_1, r_2, \ldots, r_s) \in G_s$, $\phi(n)$ is the deviation of $P_n(k) = \{(\{r_1 \times k\}, \ldots, \{r_s \times k\})$, $k = 1, 2, \ldots, n\}$ satisfy $\phi(n) = C(r, \varepsilon) n^{-1+\varepsilon}$. $C(r, \varepsilon)$ is a constant relating to r, $\varepsilon(\varepsilon > 0)$, $P_n(k)$ is called as Good-Point Set, r is the Good-Point [9].

In order to prevent the distribution of initial population from lacking of randomness, this paper append random correction to realize local randomness, it can be modelled as:

$$\overrightarrow{X} = (r_i \cdot (ub - lb) + lb) + (ub - lb) / sn \cdot 0.2 \cdot (2 \cdot rr - 1) \quad \textbf{(6)}$$

Where sn is number of population, rr is a random number in the interval [0, 1].

For example, benchmark test functions “DE JONG FUNCTION N.5”, dimension is 2, set population as 60. The initial population generated by Good-Point Set and random number method are shown in Fig. 1. The marks in left picture produced by Good-Point Set are more uniform than results of right one generated by random way.

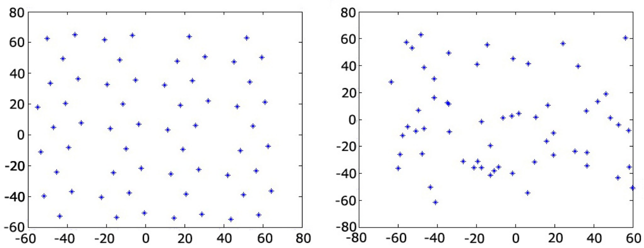


Fig. 1. Initial population from Good-Point Set and random number

3.2 Identification of Convergence State

In this section, the classification and illustration of convergence state are first provided. The Mathematical model of slope of fitting straight line and rules of classification are then proposed.

From whole picture of results after several iterations, situation of convergence curve and evolution of the best record can be classified as 3 cases: Rapid Drop, Slow Descent, Horizontal Line, descriptions are shown in Table 1.

Table 1. State of convergence curve

	Convergence State	Descriptions	Angle
0	Rapid Drop	Each iteration gets a much better record than previous iteration.	90°~135°
1	Slow Descent	Each iteration gets a better record, but tread is more and more moderate. There is a possibility of falling into local optimum.	136°~170°
2	Horizontal Line	Record of iteration is as same as previous one. Search is fallen into local optimum or there is really no better soluation.	171°~180°

About convergence curve, the latest 8 records will be kept and used to create fitting straight line for identifying the trend of convergence.

(1) Set points $(x_i, y_i), 1 \le i \le 8$ on the curve $y = \phi(x)$, find stratight line $y = f(x)$ that makes the sum of squared deviations between $y = \phi(x)$ and $y = f(x)$ is minimum.

(2) Set fitting polynomials is as $y = ax + b$, sum of squared deviations is:

$$e^2 = \sum_{i=1}^{n}\left(y_i - (ax_i + b)\right)^2 \quad \textbf{(7)}$$

(3) Calculate partial derivation of coefficient a and b respectively in Equation (7), and make value of partial derivation equal to 0.

$$\begin{cases} \dfrac{\partial e}{\partial a} = \sum_{i=1}^{n}\left(ax_i^2 + bx_i - y_i x_i\right) = 0 \\ \dfrac{\partial e}{\partial b} = \sum_{i=1}^{n}\left(ax_i + b - y_i\right) = 0 \end{cases} \quad \textbf{(8)}$$

(4) From Equation (8), a and b can be modelled as:

$$\begin{cases} a = \dfrac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \\ b = \dfrac{\sum_{i=1}^{n} y \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \end{cases} \quad \textbf{(9)}$$

(5) Through Equation (9), the slope and angle of fitting straight line can be acquired. The tread of convergence curve can describe by the slope of fitting straight lines in column "Angle" of Table 1.

For Fig. 2, the part of Convergence curve marked by 3 arrowheads. The left one is "Rapid Drop", the middle one is "Slow Descent", the right one is "Horizontal Line". Through analyzing the changing trend of convergence curve, the possibility of falling into local optimum can be handled previously, it is good for getting better solution in the end.

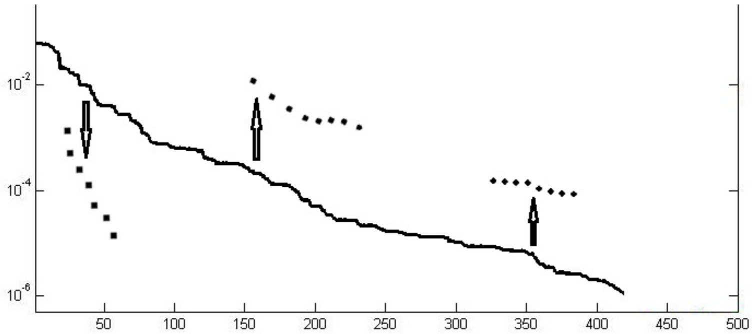


Fig. 2. Convergence curve of benchmark test function

3.3 Nonlinear Variable Convergence Factor

In this section, the definition and mathematical model of nonlinear variable convergence factor is first provided. Then characteristics of different factor at every phase in whole process of iterations are illustrated.

In original WOA, coefficient $\vec{A} \in [-a, a]$ in Equation (2) is used to balance global exploration and local exploitation. It is difficult for $\vec{a}$ as linearly factor to deal with complicated issue in reality. This paper proposes a kind of nonlinear variable convergence factor. It comprises 3 factors with different nonlinear characteristic curve, every factor decreases from 2 to 0 with respective change-pattern. The formulas of 3 convergence factors are as follow:

$$\overrightarrow{a1} = 2 - t \cdot \left(\frac{2}{T_{max}} \right) \quad \textbf{(10)}$$

$$\overrightarrow{a2} = 2 - 2\left(\frac{e^{\frac{t}{T_{max}}} - 1}{e - 1} \right)^{0.3} \quad \textbf{(11)}\ [10]$$

$$\overrightarrow{a3} = 2 \cdot \cos\left(0.5 \cdot \left(\frac{\pi \cdot t}{T_{max}} + \pi\right)\right) + 2 \quad \textbf{(12)}$$

There are 3 lines in Fig. 3. Dot-line is $\overrightarrow{a1}$ modelled as Equation (10) that is as same as $\vec{a}$ in original WOA. Dash-line is $\overrightarrow{a2}$ defined as Equation (11), situation of $\overrightarrow{a2} > 1$ accounts for a smaller proportion in whole iterations, ability of local exploitation is enhanced. On the contrary, solid-line is $\overrightarrow{a3}$ in Equation (12), situation of $\overrightarrow{a3} > 1$ take over a large proportion, global exploration ability is improved.

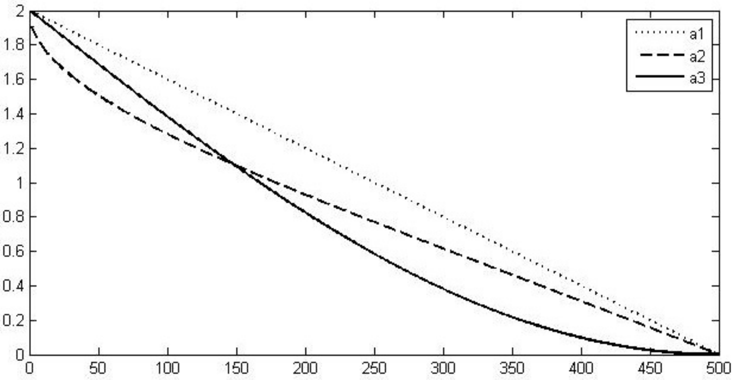


Fig. 3. Different convergence factor

Which factor is in charge to affect search is depend on the state of convergence illustrated in previous section. When Convergence State is "Rapid Drop", set $\vec{a} = \overrightarrow{a2}$ in Equation (2) to calculate $|\vec{A}|$. If $|\vec{A}| \geq 1$, global search is carry out. If $|\vec{A}| < 1$, local search execute as follow rule:

$$\overrightarrow{X1} = \overrightarrow{X^*}(t) - (2 \cdot \overrightarrow{a1} \cdot \vec{r} - \overrightarrow{a1}) \cdot \vec{D}$$
$$\overrightarrow{X2} = \overrightarrow{X^*}(t) - (2 \cdot \overrightarrow{a2} \cdot \vec{r} - \overrightarrow{a2}) \cdot \vec{D}$$
$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X1}, fobj\left(\overrightarrow{X1}\right) < fobj\left(\overrightarrow{X2}\right) \\ \overrightarrow{X2}, fobj\left(\overrightarrow{X1}\right) \geq fobj\left(\overrightarrow{X2}\right) \end{cases} \quad \textbf{(13)}$$

In Equation (13), $fobj()$ is Cost Function. Through comparing $fobj(\overrightarrow{X1})$ and $fobj(\overrightarrow{X2})$, either $\overrightarrow{X1}$ or $\overrightarrow{X2}$ derived from coefficient vectors $\overrightarrow{A1}$ and $\overrightarrow{A2}$ respectively will be assigned to $\vec{X}(t+1)$ for next iteration. If Convergence State is not "Rapid Drop", set $\vec{a} = \overrightarrow{a3}$ in Equation (2).

Hence, face to different condition in search, convergence factor is variable in 3 patterns and can apply suitable value to satisfy exploring the optimal solution.

3.4 Forced Global Search (FGS)

This section explains the solution for situation of being fallen into or having potential for local optimum, mathematical models are provided.

Original WOA algorithm takes random global search in the early stage, as with progress of iterations, all search agents move forward to the best solution obtained so far step by step. As a result, it is probable that all search agents are located in vicinity of one same local optimum, and the optimal position is no longer changed or change is too minor to take effect. The probability of falling into the local optimal solution is increased.

VCFWOA presents a method that is to force search to execute exploration and ignoring the value of $|\vec{A}|$ to improve the ability of jumping out of the local optimum. In order to control the time when search agents take into Forced Global Search, a threshold parameter is defined as follow:

Set

$$FGS\_Threshold^2 = 2p\left(t-h\right)$$

Let

$$\begin{cases} FGS\_Threshold = 0.2,\ t=1 \\ FGS\_Threshold = 0.9,\ t=T_{max} \end{cases}$$

Get

$$FGS\_Threshold = \frac{7}{50\cdot\left(T_{max}-1\right)}\cdot\left(t-\frac{81-4\cdot T_{max}}{77}\right) \tag{14}$$

Equation (14) is shown as parabola that is a nonlinear increasing function with decreasing growth rate in Fig.4 ascending from 0.2 to 0.9 within progress of 500 iterations. Cooperating with rFGS as random value in the interval [0, 1], when $rFGS < FGS\_Threshold$, search will be in global scope.

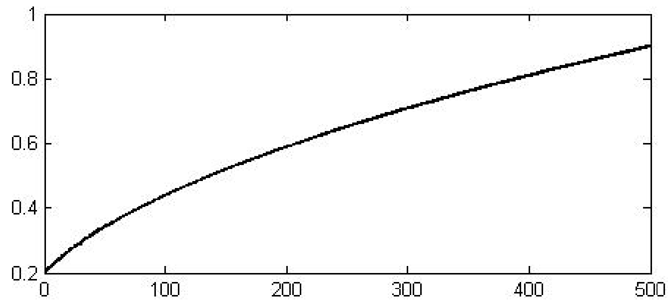


Fig. 4. Curve of threshold of FGS

In early stage of search, when Convergence State is not "Rapid Drop", so there is a strong possibility of falling in local optimum, relatively small $FGS\_Threshold$ makes FGS get involved aggressively. On the contrast, in later stage, threshold parameter is much bigger, even Convergence State is "Horizontal Line", the possibility of FGS taking effect is relevant small. When Convergence State is "Slow Descent", Set:

$$\vec{X}\left(t+1\right) = \overrightarrow{X_{rand}} - \vec{A}\cdot\left|4\cdot FGS\_Threshold\cdot\vec{C}\cdot\overrightarrow{X_{rand}} - \vec{X}\right| \tag{15}$$

When Convergence State is "Horizontal Line", Set:

$$D_{ub} = X\left(t\right) - UB$$
$$D_{lb} = X\left(t\right) - LB$$

D_{ub} and D_{lb} are distance from current whale to upper and lower boundary respectively. $D_{lb} < D_{ub}$ means that current search agents close to lower boundary, so it need to move to upper boundary by a large margin. When $D_{lb} \geq D_{ub}$, the situation is in contrary.

$$\begin{cases} \vec{X}(t+1) = \overrightarrow{X^*}(t) + FGS\_Threshold \cdot rCoefficient \cdot D_{ub},\ D_{lb} < D_{ub} \\ \vec{X}(t+1) = \overrightarrow{X^*}(t) - FGS\_Threshold \cdot rCoefficient \cdot D_{lb},\ D_{lb} \geq D_{ub} \end{cases} \quad \textbf{(16)}$$

Where $rCoefficient$ is random coefficient in the interval [0, 1].

3.5 Pseudo-code of the VCFWOA Algorithm

This section provides the complete Pseudo-code of the VCFWOA algorithm to express realization as codes of Good-Point set, Nolinear variable convergence factor as well as Forced Global Search. The Pseudo-code is as follow:

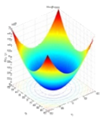
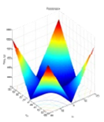
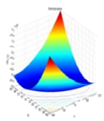
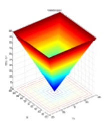
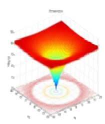
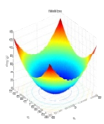
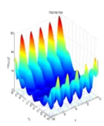
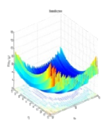
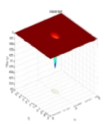
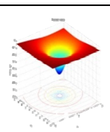
```
Initialize the whales population Xi (i=1,2,..., n)
while (t<maximum number of iterations)
  for each search agent
    Update a, a1, a2, a3, A, C, l, p, HuntStep, ConvergenceState,
    FGS_Threshold
    if1 (p<0.5)
      if4 (Convergence State in [Slow Descent, Horizontal Line])
        if3 (rFGS < FGS_Threshold)
          if5 (Convergence State = Horizontal Line)
            Update the position of the current search agent by the
            Eq.12
          end if5
          if6 (Convergence State = Slow Descent)
            Select a random search agent (X_rand)
            Update the position of the current search agent by the
            Eq.11
          end if6
        else if3
          Update the position of the current search agent by the Eq.4
        end if3
      else if4
        if2 (|A| < 1)
          Update the position of the current search agent by the Eq.8
        else if2 (|A| ≥1)
          Select a random search agent (X_rand)
          Update the position of the current search agent by the Eq.1
        end if2
      end if4
    else if1 (p≥0.5)
      Update the position of the current search by the Eq. 5
    end if1
  end for
  Calculate the fitness of each search agent
  Update X* if there is a better solution
  t=t+1
end while
return X*
```

4 Testing and Discussion

This section points out benchmark functions applied for simulation, results of experiments are provided, avantages of VCFWOA researched in this paper are presented.

The performance is tested by 10 classical benchmark functions divided into 3 groups: unimodal, multimodal, fixed-dimension multimodal. V_no indicates the number of variables, range of variables and the optimal value f_{min} are quoted in Table 2. Functions F1-F4 are unimodal with only one global optimum, which evaluate the exploitation capability of algorithms. Functions F5-F10 are multimodal functions include many local optima, they are to evaluate the exploration capability. Testing times for all function is 30, population size and maximum iteration equal to 30 and 500 have been utilized. The results are shown in Table 3 and Table 4, convergence curves are in Fig. 5 to Fig. 7.

Table 2. Description of benchmark functions

Function		V_no	Range	f_{min}
	$F_1(x)=\sum_{i=1}^{n} x_i^2$	30	[-100, 100]	0
	$F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[-10, 10]	0
	$F_3(x)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i} x_j\right)^2$	30	[-100, 100]	0
	$F_4(x)=\max_i\left\{\lvert x_i\rvert, 1\le i\le n\right\}$	30	[-100, 100]	0
	$F_5(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2})$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	30	[-32, 32]	0
	$F_6(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600, 600]	0
	$F_7(x)=\frac{\pi}{n}\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2$ $[1+10sin^2(\pi y_{i+1})]+(y_n-1)^2\}+\sum_{i=1}^{n}u(x_i,10,100,4)$ $y_i=1+\frac{x_i+1}{4}\quad u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, x_i>a\\ 0, -a<x_i<a\\ k(-x_i-a)^m, x_i<-a\end{cases}$	30	[-50, 50]	0
	$F_8(x)=0.1\{sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2\left[1+sin^2(3\pi x_i+1)\right]$ $+(x_n-1)^2\left[1+sin^2(2\pi x_n)\right]\}+\sum_{i=1}^{n}u(x_i,5,100,4)$	30	[-50, 50]	0
	$F_9(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right)$	3	[1, 3]	-3.86
	$F_{10}(x)=-\sum_{i=1}^{5}\left[(X-a_i)(X-a_i)^T+C_i\right]^{-1}$	4	[0, 10]	−10.1532

4.1 Comparison between VCFWOA and Meta-heuristic Optimization Algorithm

This section takes comparison VCFWOA to classical meta-heuristic optimization algorithm WOA, PSO, GSA and DE to analyze performance. The mean, standard deviation, best value and worst value are shown in Table 3.

Table 3. Comparison between VCFWOA and Meta-heuristic optimization algorithm

		WOA	PSO	GSA	DE	VCFWOA
f1	mean	1.4821E-72	1.7603E-04	3.2095E-16	1.0087E-13	1.7566E-126
	std	7.8789E-72	2.5847E-04	1.2528E-16	7.6057E-14	6.6141E-126
	best	8.2759E-89	1.3503E-04	2.3394E-16	9.7684E-14	6.7701E-140
	worst	4.3189E-71	1.8794E-04	4.1666E-16	1.2878E-13	3.1840E-125
f2	mean	5.3295E-49	5.0060E-02	6.5179E-02	1.9445E-09	6.9376E-79
	std	2.8669E-48	5.3172E-02	2.4906E-01	1.1976E-09	2.9097E-78
	best	1.0412E-56	3.7084E-02	6.0565E-02	1.8945E-09	5.1267E-88
	worst	1.5711E-47	5.1015E-02	7.6881E-02	2.0815E-09	1.5787E-77
f3	mean	4.4231E+04	8.7024E+01	1.0891E+03	8.4952E-11	1.4233E+01
	std	1.3149E+04	2.5739E+01	3.7893E+02	9.1352E-11	7.4882E+01
	best	2.2000E+04	7.8582E+01	7.6793E+02	6.7615E-11	2.4098E-23
	worst	6.7400E+04	9.3767E+01	1.2603E+03	9.0937E-11	4.1050E+02
f4	mean	4.9334E+01	1.3507	8.5485	0	5.6491E-19
	std	2.8664E+01	4.0925E-01	2.1532	0	1.9737E-18
	best	2.5710	1.3262	7.5307	0	1.0680E-31
	worst	8.7202E+01	1.4830	9.2821	0	8.2152E-18
f5	mean	3.9672E-15	3.5833E-01	7.9238E-02	1.1773E-07	8.8818E-16
	std	2.7572E-15	6.2998E-01	2.7623E-01	4.9634E-08	0
	best	8.8818E-16	3.0630E-01	7.1363E-02	1.1243E-07	8.8818E-16
	worst	7.9936E-15	3.9620E-01	9.1846E-02	1.4615E-07	8.8818E-16
f6	mean	3.1556E-03	1.1508E-02	3.2433E+01	0	0
	std	1.7284E-02	9.7307E-03	6.4947	0	0
	best	0	9.7047E-03	3.1071E+01	0	0
	worst	9.4669E-02	1.3408E-02	3.7957E+01	0	0
f7	mean	1.8106E-02	8.6837E-03	2.3097	9.7585E-15	2.8830E-06
	std	1.1482E-02	3.3169E-02	1.2161	9.8451E-15	1.6534E-06
	best	5.4509E-03	7.1818E-03	1.9013	7.7959E-15	4.8482E-07
	worst	5.7491E-02	1.0296E-02	2.8486	1.2670E-14	7.1491E-06
f8	mean	5.5415E-01	8.0768E-03	1.1502E+01	6.0153E-14	1.0366E-01
	std	3.0836E-01	1.1258E-02	8.6070	5.9933E-14	1.3601E-01
	best	5.4699E-02	6.6222E-03	9.8519	5.1659E-14	1.0094E-06
	worst	1.2414	9.8913E-03	1.2010E+01	7.4832E-14	5.7797E-01
f9	mean	-3.8517	-4.9838	-4.5703	0	-3.8624
	std	3.2977E-02	3.2196E-15	2.7536E-15	0	1.0154E-03
	best	-3.8628	-4.8229	-4.2408	0	-3.8628
	worst	-3.6814	-5.9523	-5.1271	0	-3.8576
f10	mean	-8.4489	-8.3051	-6.9363	-1.1738E+01	-9.1336
	std	2.6795	3.6706	4.7843	2.8907E-06	2.0741
	best	-1.0152E+01	-7.3966	-6.3647	-1.1006E+01	-1.0153E+01
	worst	-2.6299	-8.7546	-7.5271	-1.2112E+01	-5.0552

As shown in Table 3, for 10 benchmark functions, according to mean and standard deviation of best solution, the accuracy and robustness of VCFWOA in this paper is very competitive with any other 3 algorithms. Except for F4, DE occupies a position of prominence.

4.2 Comparison between VCFWOA and Other Improved WOA

This section takes comparison VCFWOA to other improved WOA algorithm AWOA, CSWOA, DEOBWOA to analyze performance. Testing results are shown in Table 4.

Table 4. Comparison between VCFWOA and other improved WOA

		AWOA	CSCWOA	DEOBWOA	VCFWOA
f1	mean	1.9921E-93	2.1576E-101	3.6068E-114	1.7566E-126
	std	9.1977E-98	3.7309E-87	9.0790E-105	6.6141E-126
	best	1.9385E-93	2.1066E-101	3.5296E-114	6.7701E-140
	worst	2.0076E-93	2.1947E-101	3.7149E-114	3.1840E-125
f2	mean	5.2481E-69	1.9532E-70	3.9376E-77	6.9376E-79
	std	3.4807E-72	3.0303E-57	6.5314E-61	2.9097E-78
	best	5.1063E-69	1.9170E-70	3.8779E-77	5.1267E-88
	worst	5.2985E-69	2.0021E-70	3.9828E-77	1.5787E-77
f3	mean	2.5765E+01	4.5323E+02	1.5632E+01	1.4233E+01
	std	4.4859E+02	2.6664E+02	8.8325E+03	7.4882E+01
	best	2.5390E+01	4.4084E+02	1.5266E+01	2.4098E-23
	worst	2.5897E+01	4.6419E+02	1.5784E+01	4.1050E+02
f4	mean	4.3386E+01	4.5120E+01	4.4401E+01	5.6491E-19
	std	6.5162E-06	8.0486E-14	1.1288E-02	1.9737E-18
	best	3.2158E+01	3.3843E+01	4.3171E+01	1.0680E-31
	worst	4.4279E+01	4.6330E+01	4.4578E+01	8.2152E-18
f5	mean	3.5823E-15	7.0462E-15	2.7238E-16	8.8818E-16
	std	1.3481E-16	0	2.0187E-15	0
	best	3.5629E-15	8.8714E-16	2.7051E-16	8.8818E-16
	worst	3.6412E-15	7.0649E-15	2.7735E-15	8.8818E-16
f6	mean	1.3420E-03	2.2313E-03	0	0
	std	0	2.9174E-01	0	0
	best	1.3268E-03	2.1784E-03	1.2357E-03	0
	worst	1.3638E-03	2.2532E-03	1.2660E-03	0
f7	mean	4.3476E-03	7.2441E-03	3.7830E-06	2.8830E-06
	std	9.6811E-03	1.4173E-05	1.0377E-03	1.6534E-06
	best	4.2948E-03	7.1397E-03	3.7228E-06	4.8482E-07
	worst	4.3585E-03	7.2865E-03	3.8199E-06	7.1491E-06
f8	mean	2.8386E-01	3.2891E-01	4.4153E-01	1.0366E-01
	std	2.4160E-01	1.3867E-01	2.4160E-01	1.3601E-01
	best	2.8302E-01	3.2573E-02	4.3690E-04	1.0094E-06
	worst	6.4878E-01	3.3132E-01	5.7932	5.7797E-01
f9	mean	-3.8541	-3.8702	-3.8522	-3.8624
	std	1.1593E-02	2.4013E-02	1.1593E-02	1.0154E-03
	best	-3.8179	-3.7796	-3.8020	-3.8628
	worst	-3.5647	-2.0916	-3.7453	-3.8576
f10	mean	-8.6702	-8.8836	-8.2149	-9.1336
	std	2.0499	2.1459	2.0069	2.0341
	best	-8.5213	-8.7553	-8.1788	-1.0153E+01
	worst	-8.7434	-4.1404	-8.2467	-5.0552

As shown in Table 4, VCFWOA have advantages on mean, standard deviation, best value and worst value over AWOA and CSCWOA. In particular, it was more efficient for functions F1 and F2. Also, the results reported in for functions F4 indicate that VCFWOA has better exploration capability, except for F3 of DEOBWOA, but there is little difference in order of magnitude. For best solution of F2, VCFWOA get more than one order of magnitude higher than other improved WOA.

Fig. 5 is comparison results about unimodal benchmark functions. VCFWOA in F1, F4 has the fastest convergence speed and best convergence accuracy. Especially in F4, solution of VCFWOA is much better than any others. In F3, VCFWOA go through serval times of jumping out of local optimum to get better result.

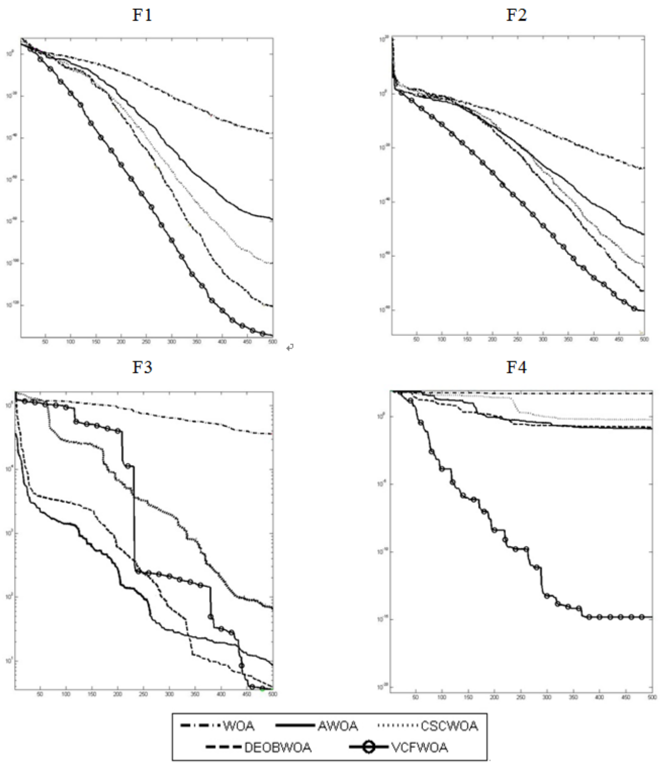


Fig. 5. Convergence curves obtained in unimodal benchmark functions

Fig. 6 is comparison of multimodal benchmark functions. In F7, even VCFWOA is slower than CSCWOA in convergence speed, it get the best solution finally. In F6 and F8, VCFWOA has the fastest convergence speed and best convergence accuracy. When original WOA is sinking into local optimum and difficult to find better solution, VCFWOA can jump out it rapidly by analyzing tread of convergence state and executing forced global search.

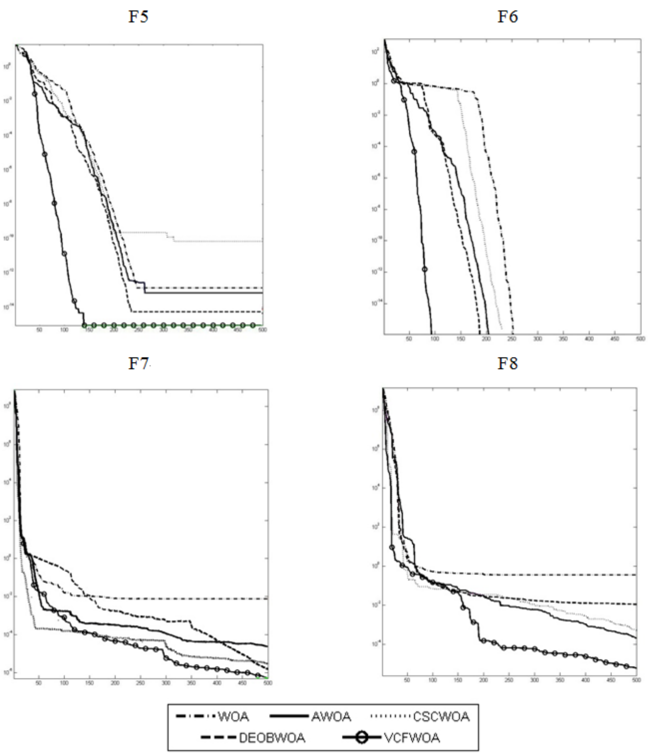


Fig. 6. Convergence curves obtained in multimodal benchmark functions

In Fig. 7, all kinds of improved WOA have similar results, however, VCFWOA enjoy the fastest speed of convergence.

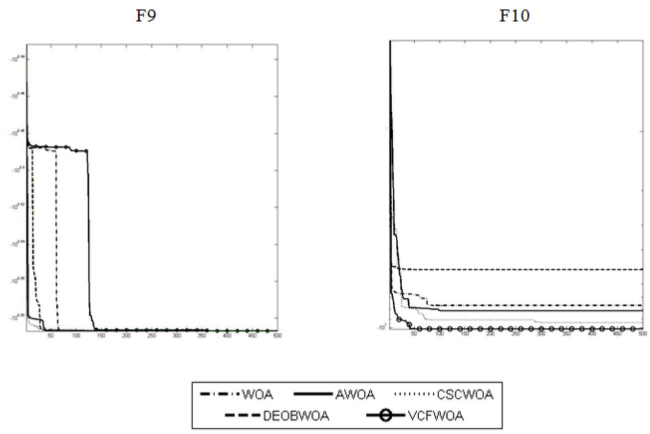


Fig. 7. Convergence curves obtained in fixed-dimenstion multimodal benchmark functions

5 Conclusions

In order to overcome the defects as well as improve the performance of original WOA, this paper presented several procedures, such as initializing population by Good-Point Set, identifying convergence state through fitting straight line. Nonlinear variable convergence factor and forced global search make search adopt well-directed update-pattern.

After tested by 10 classical benchmark functions and compared with other improved WOA, results and convergence curves indicate that exploration, exploitation, local optimum avoidance of VCFWOA are competitive, convergence speed and robustness are promoted.

Acknowledgements

This research was supported by Tianjin Natural Science Foundation (Research on key problems of source location of gas leakage in 3D complex flow field based on compressed sensing sensor network, 20JCYBJC00320)

References

[1] S. Mirjalili, A. Lewis, The whale optimization algorithm, Advances in Engineering Software 95(2016) 51-67.

[2] Z.S. Liu, S. Li, Whale Optimization Algorithm based on Chaotic Sine Cosine, Computer Engineering and Applications 54(7)(2018) 159-163.

[3] Y.J. Sun, X.L. Wang, Y.H. Chen, Z.J. Liu, A modified whale optimization algorithm for large-scale global optimization problems, Expert Systems with Applications 114(2018) 563-577.

[4] A. Mostafa, A.E. Hassanien, M. Houseni, H. Hefny, Liver segmentation in MRI images based on whale optimization algorithm, Multimedia Tools and Applications 56(4)(2017) 499-509.

[5] Y.F. Xu, F. Qian, M.L. Yang, W.L. Du, W.M. Zhong, Improved whale optimization algorithm and its application in optimization of residue hydrogenation parameters, CIESC Journal 69(3)(2018) 891-899.

[6] P.F. Niu, Z.L. Wu, Y.P. Ma, C.J. Shi, J.B. Li, Prediction of steam turbine heat consumption rate based on whale optimization algorithm, CIESC Journal 68(8)(2017) 1049-1057.

[7] F. Yu, X.Z. Xu, A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network, Applied Energy 134(2014) 102-113.

[8] L.G. Hua, Y. Wang, Applications of Number Theoretic Methods in Approximate Analysis, 1978.

[9] C.Y. Zhao, L. Zhang, Application of Good-Point Set Genetic Algorithm, Microcomputer Development 5(2000) 1-2.

[10] Y.C. Huang, L.B. Zhang, Improved whale optimization algorithm and its application, Computer Engineering and Applications 55(21)(2019) 220-226.