ISSN 1330-3651(Print), ISSN 1848-6339 (Online)
UDC/UDK 004.021:519.87

AN IMPROVED REAL HYBRID GENETIC ALGORITHM

Weidong Ji, Jianhua Wang, Jun Zhang

Original scientific paper
Aiming at the problem of premature convergence of genetic algorithm and particle swarm algorithm, in order to let the two methods converge to the global optimal solution with the greatest probability and improve the efficiency of the algorithm, the paper will combine improved genetic algorithm with improved particle swarm optimization method to constitute mixed improved algorithm. Through multiple benchmark function used to test the performance of real hybrid genetic algorithm, the results show that hybrid algorithm has good global search ability, fast convergence, good quality of the solution, and good robust performance of its optimization results.

Keywords: *genetic algorithm, particle swarm optimization, hybrid algorithm*

Poboljšani stvarni hibridni genetski algoritam

Izvorni znanstveni članak
Želeći riješiti problem prerane konvergencije genetskog algoritma i algoritma roja čestica, kako bi se omogućilo da te dvije metode konvergiraju ka globalnom optimalnom rješenju uz najveću vjerojatnoću te da se poboljša učinkovitost algoritma, u članku će se kombinirati poboljšani genetski algoritam s metodom poboljšane optimalizacije roja čestica da bi se sastavio miješani poboljšani algoritam. Uz različite referentne funkcije upotrjebljene za testiranje funkcioniranja stvarno hibridnog genetskog algoritma, rezultati pokazuju da hibridni algoritam ima dobru globalnu sposobnost pretraživanja, brzu konvergenciju, dobru kvalitetu rješenja i dobru performansu rezultata optimalizacije.

Ključne riječi: *genetski algoritam, optimalizacija roja čestica, hibridni algoritam*

1 Introduction

Genetic algorithm has been widely used in many fields [1 ÷ 4], but also there are many shortcomings and inadequacies in urgent need of improvement [5 ÷ 7], mainly in the following aspects: (1) algorithm has premature convergence problem (Premature Convergence). Compared with other methods, genetic algorithm can solve this problem well, but still not satisfactory; (2) algorithm has the problem of low convergence speed, especially when dealing with high dimension and a higher degree of complexity problem, the problem is particularly prominent. These two issues is one of the principal contradictions in the application of genetic algorithm, how to ensure the global convergence of genetic algorithm and the search efficiency of genetic algorithm at the same time is an emphasis and difficulty in the study of the theory and application of genetic algorithm.

In view of the above problems, such as type (1) with variable upper and lower constraints optimization problem, this paper adopts real-coded genetic algorithm, and start from the perspective of the overall algorithm to improve the genetic manipulation of conventional genetic algorithm.

$$\min f(x) = f(x_1, x_2, ...x_n)$$
$$x_i^{\min} \le x_i \le x_i^{\max} (i = 1, 2, ..., n). \tag{1}$$

In the type, $f(x)$ is the objective function, $x = (x_1, x_2, \cdots, x_n)$ is a vector optimization objective of the search space $\Re^n$, $x_i^{\min}$, $x_i^{\max}$ respectively as optimization variables x lower bound and upper bound, $x_i \in [x_i^{\min}, x_i^{\max}]$.

2 Improved genetic algorithm (IGA)

The optimal solution search process of genetic algorithm is to simulate Darwin's theory of evolution and "survival of the fittest" evolutionary thought. Form Eq. (1) with variable upper and lower limits of constrained optimization problem, this paper introduces ecology niche idea of its construction, so that the individual has the ability to learn a certain extent, use the real-coded genetic algorithm, and to start from the perspective of the overall algorithm to improve the optimized performance of genetic algorithm .

2.1 The chaotic generating initial population

Chaos phenomenon means to occur in a deterministic system seemingly random irregular movement, a deterministic theoretical description of the system, and its behavior was expressed as the uncertainty, unrepeatable and unpredictable, which is called chaotic phenomenon. Chaos is a kind of non-periodic order. Among them, ergodicity of chaos characteristics can be used as the search process to avoid falling into the local minimum of an optimized mechanism, and there is a clear distinction between it and simulated annealing probability inferior to the transfer. The initial population generated by the search of chaotic sequence, which to some extent can improve the search efficiency of the genetic algorithm, while increasing the diversity of the population.

Logistic map shown in the Eq. (2)

$$\alpha_i^{r+1} = \mu \alpha_i^r (1 - \alpha_i^r) \tag{2}$$

takes different values in the μ, α_i^r will appear from the bifurcation to a chaotic process, as shown in Fig. 1, f_{best}

$\alpha_i^0 = 0{,}1$. As you can see from the figure, for $\mu \geq 3{,}9$, chaos appears.

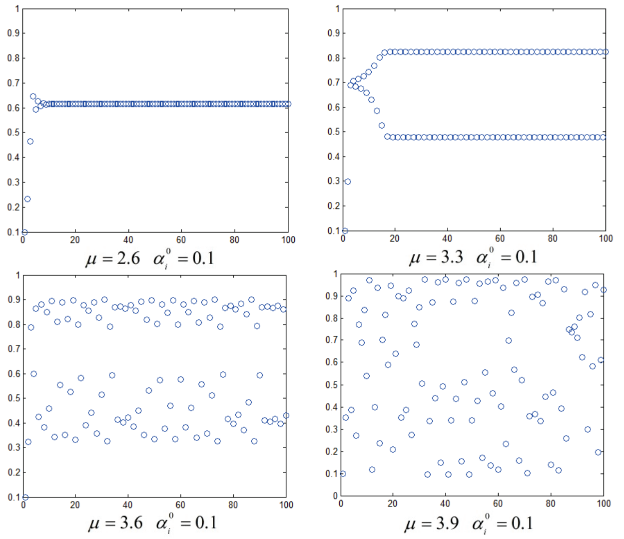


Figure 1 Bifurcation and Chaos Phenomenon of Logistic Sequence

In this paper, where the control parameters to take $\mu = 4$; i is the serial number of the chaotic variable, $i = 1, 2,..., n$; r means the population number, $r = 1, 2,..., P$; P means the population size; α_i means chaotic variables, $0 \leq \alpha_i \leq 1$. Using the sensitive characteristic of chaotic initial value, to the type (2) initial n small difference values, in order to avoid the cycle, value cannot be 0; 0,25; 0,50; 0,75; 1; so we can get n chaotic variables, α_i^1, $r = 1, 2, 3... P$, getting P initial populations.

The chaotic variables α_i^r inverse mapping to obtain the corresponding variables x_i^r in optimization space

$$x_i^r = x_i^{\min} + (x_i^{\max} - x_i^{\min})\alpha_i^r, \tag{3}$$

in the Eq. (3), the change interval of $[x_i^{\min}, x_i^{\max}]$ is x_i, $i = 1, 2,..., n$.

2.2 Non-linear ranking selection operator

The selection operator is based on the individual fitness function value to the operation process of the individual choice. Its function is choosing some more excellent individuals from the current population. The non-linear ranking selection operator can effectively maintain the iterative process of population diversity.

Sort the group whose size is P $pop = \{x_1, x_2, \cdots x_i, \cdots, x_P\}$ according to the fitness value in descending order, individual x_i's selection probability $\varphi_s(x_i)$ as shown in Eq. (4)

$$\begin{cases} \varphi_s(x_i) = q'(1-q)^{f_r(x_i)-1} \\ q' = \dfrac{q}{1-(1-q)^P} \end{cases}. \tag{4}$$

In the equation, q is the individual selected probability, $f_r(x_i)$ is the individual arrangement number.

After each chromosome selection probability is determined, we can use the proportional selection operator to select excellent individuals.

2.3 Environmental evolution crossover operator

In ecology, the environment has a relatively large impact on the learning ability of the biological and biological evolution, in a different environment, the creature has the ability to repair itself [9].

By this theory inspired, this article according to the "niche construction" introduces the environment variables, refers to the individual to repair its environment in the learning process, and then through environment variables in reaction to its genetic code, and then reflect learning ability in the process of evolution of the group. Environmental evolution crossover operator described as Eqs. (5) and (6):

$$\xi_c = \begin{cases} X_s^t + \omega_1 \gamma^t \\ X_f^t + \omega_1 \gamma^t \end{cases} \tag{5}$$

$$\gamma^t = \begin{cases} \omega_2 \gamma^{t-1} + \omega_3 (X_{\text{best}}^t - X_f^t) + \omega_4 (X_s^t - X_f^t), f(X_s^t) > f(X_f^t) \\ \omega_2 \gamma^{t-1} + \omega_3 (X_{\text{best}}^t - X_s^t) + \omega_4 (X_f^t - X_s^t), f(X_f^t) > f(X_s^t) \end{cases}, \tag{6}$$

where, ξ_c is the environmental evolution of the crossover operator, t is the evolution algebra, X_s^t and X_f^t are father generation ($s \neq f$), X_{best}^t is the fitness value for the current best individual, γ^t is an environment variable set, ω_1, ω_2, ω_3 and ω_4 are as a parameter coefficient, using dynamic value strategy, $\omega \in [0,1]$.

With the establishment of environmental evolution crossover operator, individuals of groups can keep track of the current fitness value of the best individual to carry on cross-operation, improve the efficiency of crossover operator, at the same time be able to obtain the evolutionary history of experience, to make the environment evolution crossover operator have the capacity of learning, thus making the self-learning ability in a certain extent.

2.4 Non-uniform mutation operator

During the natural evolution process, in the evolution of the initial phase there is a large variation space, but in the later stage of evolution variability space becomes smaller. Inspired by this phenomenon, this paper uses a non-uniform mutation operator to ensure that the mutation step size decreases with increasing time, with the evolution of the initial variation space and space of variation in the later stage of evolution consistent.

Let the elements x_m be the elements of variation to the variation of probability p_m which is selected from the self-chromosome $x = (x_1, x_2,..., x_m,..., x_n)$, $x_m \in \left[x_m^{\min}, x_m^{\max}\right]$, x_m^{new} is the variation element generated by Eq. (7):

$$x_m^{\text{new}} \in \Big|\max\left|x_m(t) - \Delta(t, x_m(t) - x_m^{\min}), x_m^{\min}\right|, \\ \min\left|x_m(t) + \Delta(t, x_m^{\max} - x_m(t)), x_m^{\max}\right|\Big| . \tag{7}$$

$$\Delta(t,d)=d(1-r^{(1-t/n_t)^5}) \tag{8}$$

In the above formula, it generates x_m^{new} instead of x_m within the range randomly.

Type (8), r refers to the uniform distribution on the interval (0,1), n_t is the maximum number of iterations, t is the evolution algebra. From Eq. (8) can be seen when the evolution algebra, t increases, $\Delta(t, d)$ value decreased gradually in the formula, in the initial stage of evolution, the r value is smaller, the $\Delta(t, d)$ value close to 1, the spatial variation is large; while in the later stage of evolution, when t and n_t approache, $\Delta(t, d)$ is close to 0, the spatial variation is smaller. Characteristics of the non-uniform mutation operator is a $\Delta(t, d)$ returns a value in the range of [0, d], and $\lim\limits_{t\to+\infty}\Delta(t,d)=0$, to ensure that the mutation step size decreased with increasing time of t, consistent in the initial stage of evolution of spatial variability and spatial variability in the later evolution of small.

3 Improved particle swarm optimization

3.1 The introduction of the algorithm

Numerous studies indicate that the Basic PSO model is easy Poa [11 ÷ 13], convergence of the position and velocity of the particle is not changed or changes only slightly. One reason for fast convergence lies in the standard PSO iteration, each particle swarm of respectively from two extreme p_{best} and g_{best} obtained in "individual recognition" and "social cognitive ability", under this influence the change of particle's flight speed and position. In a complex environment with a large number of local extreme values, g_{best} is easy to fall into local extremum, and the PSO algorithm itself does not provide any mechanism that allows g_{best} to escape from the local minimum, and then the PSO algorithm premature convergence. Aiming at the above problem, cloning theory is used in artificial immune system to avoid premature convergence of the algorithm.

3.2 Cloning mutation operator

The artificial immune system (Artificial Immune System, AIS) is an intelligent method to mimic the natural immune system function [14]. Cloning is the English the Clone term single translation, meaning clones, continuous passage through asexual reproduction (such as cell mitosis) and the formation of groups, a description commonly used at the cellular level.

Inspired by the theory, the paper designed clonal mutation operator to change the position of the particle populations vector, and thus avoid falling into local minima, to better improve the value of the particle fitness. Cloning mutation operator is specifically described as follows: Cloning mutation operator for each component of the position vector with a random value.

$$x_{ij}(t)=\begin{cases} x'_{ij}(t)+\Delta(t,x_{\max,j}-x'_{ij}(t)) & \text{if } U(0,1)\le 0{,}5 \\ x'_{ij}(t)-\Delta(t,x'_{ij}(t)-x_{\min,j}) & \text{else} \end{cases} . \tag{9}$$

Here, $x'_{ij}(t)$ is the value after the cloning $x_{ij}(t)$, $\Delta(t,d)=d(1-r^{(1-t/n_t)^5})$ with type (8).

Particle swarm optimization algorithm, each iteration of each particle of the population asexual reproduction, the formation of a scale and nature of the original group of clone groups, each particle of the clone groups mutation operation to calculate the cloning of particle value of fitness, if it is higher than the particles, particles of itself will be replaced, otherwise the position of the particle is the same.

With clonal variation, each particle of the population has a self-learning capability enabling the particles to enhance the ability to get rid of local minima and improving the efficiency of the algorithm.

Improved particle swarm optimization algorithm (10) (11) updates on the velocity and position update.

$$v_{ij}(t+1)=\chi[v_{ij}(t)+\phi_1 r_1(p_{\text{best}(i,j)}(t)-x_{ij}(t))+ \\ +\phi_2 r_2(g_{\text{best}(j)}(t)-x_{ij}(t))] \tag{10}$$

$$x_{i,j}(t+1)=x_{i,j}(t)+v_{i,j}(t+1) \tag{11}$$

where $\chi=\dfrac{2}{\left|2-\phi-\sqrt{\phi(\phi-4)}\right|}$, $\phi=\phi_1+\phi_2$. (12)

4 The improved hybrid genetic algorithm

4.1 The description of improved hybrid genetic algorithm

Aiming at the problem of premature convergence of genetic algorithm and particle swarm algorithm are present, so that the two methods can converge to the global optimal solution more probability and improve the efficiency of the algorithm, the genetic algorithm and particle swarm optimization method is improved by combining hybrid algorithm, as follows:

Set the population size is P, the population of all individuals according to the fitness value ranking, divide it into 2 parts, to generate new population according to the interaction between the 2 strategies of the following:

(1) The improved PSO optimization strategy: the improved adaptive PSO algorithm M individuals using the improved value of optimal scheduling after the update, the update process memory each particle p_{best}, finally by p_{best} instead of particles, and then generate the M progeny.

(2) The improved GA optimization strategy: population remaining in the individual $P-M$, use the GA optimized.

From the description of the algorithm, we can see that when M = 0, the hybrid algorithm has become this improved genetic algorithm (IGA), when M = 0,5P, hybrid algorithm every iterative evolution individuals generated by the improved particle swarm optimization M is 50 % of the total number of groups P algorithm mixed with M in the population number P in the proportion of M/P to quantify, said hybrid algorithm is generally depending on the problem to select the appropriate value of the degree of mixing, the mixed is generally selected at around 20 %.

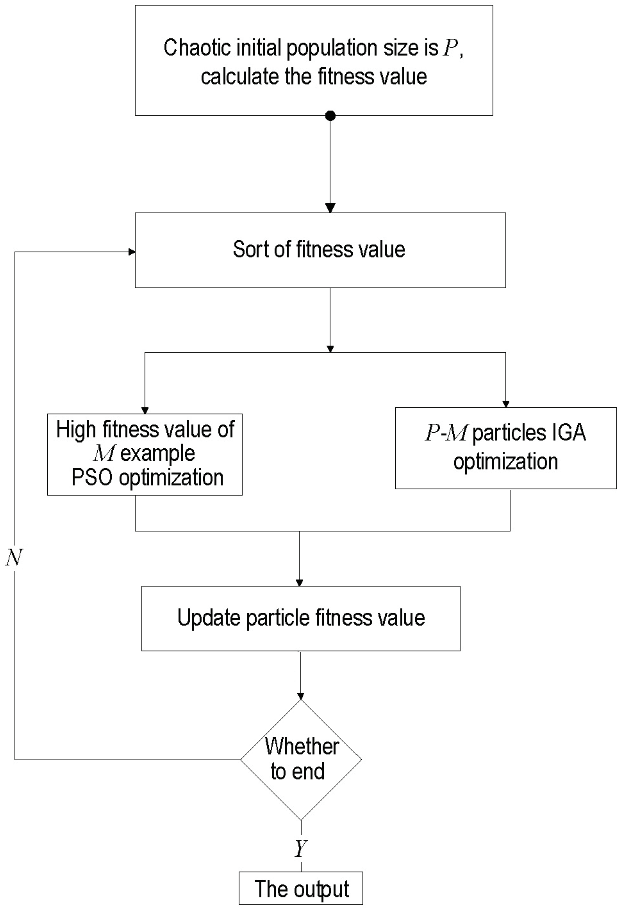


Figure 2 Flowchart of improved hybrid genetic algorithm

4.2 The progress of the improved hybrid genetic algorithm

On the basis of the above theory, this paper improved hybrid genetic algorithm process can be summarized in the following steps:

(1) Chaotic mapping generates the initial population and calculates the population fitness value.
(2) Sort the fitness value of the individuals in a population.
(3) The improved particle swarm optimization strategy, the M individuals with high fitness value, the particle swarm optimization, with each individual p_{best} instead of individual itself, to produce M individuals.
(4) Improved genetic algorithms strategy operates on the remaining $M–P$ individuals.
(5) Re-calculate the fitness value
(6) If the hybrid algorithm termination condition is reached, show the output optimal results and the iteration is over, otherwise go to step (2) to continue the evolutionary search.

Detailed flow as shown in Fig. 2.

5 Numerical experiments and results

5.1 Comparative test of the algorithm performance among IHGA, SGA and IGA

Using Tab. 1 benchmark functions to test the algorithm optimization [15], benchmark function detail see references [15] .

Table 1 Benchmark test functions

Test functions	Dimensionality	The feasible solution space
$f_1(x)=\sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^n$
$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	$[-10, 10]^n$
$f_3(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	30	$[-100, 100]^n$
$f_4(x)=\max\left\{\lvert x_i\rvert,\ i=1,2,\cdots,n\right\}$	30	$[-100, 100]^n$
$f_5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	$[-30, 30]^n$
$f_6(x)=\sum_{i=1}^{n}\left(-x_i \sin\sqrt{\lvert x_i\rvert}\right)$	30	$[-500, 500]^n$
$f_7(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	30	$[-5,12; 5,12]^n$
$f_8(x)=-20\exp\left(-0{,}2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+\exp(1)$	30	$[-32, 32]^n$
$f_9(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	30	$[-600, 600]^n$
$f_{10}(x)=\frac{\pi}{n}\left\{10\sin^2(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\right\}+$ $+\sum_{i=1}^{n}u(x_i,10,100,4)$	30	$[-50, 50]^n$

$f_{11}(x)=4x_1^2-2,1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	$[-5,5]^n$
$f_{12}(x)=\left(x_2-(5,1/4\pi^2)x_1^2+(5/\pi)x_1-6\right)^2+10\left(1-(1/8\pi)\right)\cos x_1+10$	2	$[-5,10]\times[0,15]$
$f_{13}(x)=-\sum_{i=1}^{10}\left[(\boldsymbol{x}-a_i)(\boldsymbol{x}-a_i)^T+c_i\right]^{-1}$	4	$[0,10]^n$
$f_{14}(x)=\frac{1}{10}\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})]+ +(x_n-1)^2[1+\sin^2(2\pi x_n)]\right\}+\sum_{i=1}^{n}u(x_i,5,100,4)$	30	$[-50,50]^n$

Table 2 Number of generations for test functions

Test function	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
Times of iterations	400	500	400	500	400	500	30	50	40	500	60	60	60	500

Table 3 Simulation results

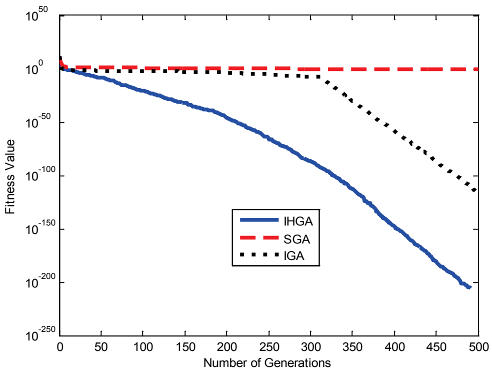
Test function	Average fitness value(Standard deviation)			Global minimum
	IHGA	IGA	SGA	
f_1	0 (0)	1,0179E-182 (0)	5,4215 (1,0169)	0
f_2	0 (0)	1,3250E-116 (4,4662E-116)	2,5714E-2 (6,7813E-2)	0
f_3	0 (0)	2,5322E-187 (0)	891,7700 (222,33)	0
f_4	0 (0)	4,6758E-113 (2,5382E-112)	1,3622 (1,4465E-1)	0
f_5	2,7663E-2 (2,5752E-2)	2,3372E-1 (2,2027E-1)	257,2472 (63,1038)	0
f_6	−12569,4690 (1,0604E-3)	−12569,4550 (1,5434E-2)	−6120,3977 (483,2116)	−12569,5
f_7	0 (0)	7,0806E-14 (4,3697E-13)	164,3816 (11,3370)	0
f_8	4,4409E-16 (0)	1,1685E-13 (2,1318E-13)	5,9202 (6,4707E-1)	0
f_9	0 (0)	3,0183E-14 (1,5474E-13)	4,8696 (1,0262)	0
f_{10}	3,9470E-6 (3,3575E-6)	1,8851E-5 (9,3391E-6)	1,2586E-2 (3,7359E-3)	0
f_{11}	−1,0316280 (1,9361E-6)	−1,0314189 (8,9109E-4)	−1,0311546 (1,0881E-3)	−1,0316285
f_{12}	0,39788748 (8,1303E-7)	0,39926486 (6,0594E-3)	0,40188725 (1,7276E-2)	0,39788736
f_{13}	−10,533990 (5,6384E-3)	−10,532968 (5,5076E-3)	−6,8929414 (3,2702)	−10,536284
f_{14}	9,9978E-5 (8,5678E-5)	4,9554E-4 (1,5758E-3)	4,4875 E-1 (1,2718E-1)	0

Table 4 Performance comparison among three algorithms

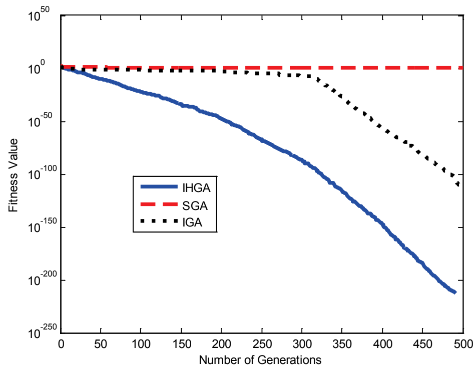
Test function	The fitness value function to calculate the number of times			The average fitness value (the fitness value of standard deviation)			Global minimum
	IHGA	OGA/Q	FEP	IHGA	OGA/Q	FEP	
f_1	54 683	112 559	150 000	0 (0)	0 (0)	5,7E-4 (1,3E-4)	0
f_2	68 369	112 612	200 000	0 (0)	0 (0)	8,1E-3 (7,7E-4)	0
f_3	54 702	112 576	500 000	0 (0)	0 (0)	1,6E-2 (1,4E-2)	0
f_4	68 434	112 893	500 000	0 (0)	0 (0)	0,3 (0,5)	0
f_5	54 785	—	2 000 000	2,7663E-2 (2,5752E-2)	—	5,06 (5,87)	0
f_6	68 440	302 166	900 000	−12569,4690 (1,0604E-3)	−12569,4537 (6,4470E-4)	−12554,5 (52,6)	−12569,5

Table 4 Performance comparison among three algorithms (continued)

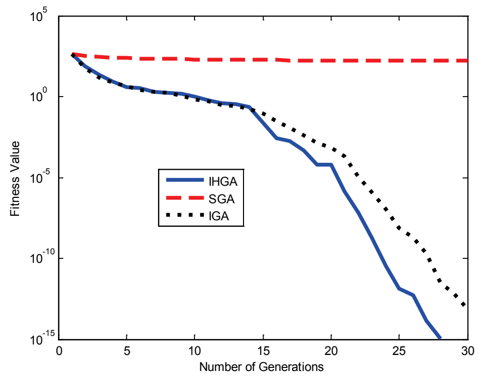
Test function	The fitness value function to calculate the number of times			The average fitness value (the fitness value of standard deviation)			Global minimum
	IHGA	OGA/Q	FEP	IHGA	OGA/Q	FEP	
f_7	4 113	224 710	500 000	0 (0)	0 (0)	4,6E-2 (1,2E-2)	0
f_8	6 864	112 421	150 000	4,4409E-16 (0)	4,4400E-16 (3,9890E-17)	1,8E-2 (2,1E-3)	0
f_9	5 484	134 000	200 000	0 (0)	0 (0)	1,6E-2 (2,2E-3)	0
f_{10}	68 379	134 556	150 000	3,9470E-6 (3,3575E-6)	6,0190E-6 (1,1590E-6)	9,2E-6 (3,6E-6)	0
f_{11}	8 219	—	10 000	−1,0316280 (1,9361E-6)	—	−1,03 (4,9E-7)	−1,0316285
f_{12}	8 224	—	10 000	0,39788748 (8,1303E-7)	—	0,3980 (1,5E-7)	0,39788736
f_{13}	8 224	—	10 000	−10,533990 (5,6384E-3)	—	−6,57 (3,14)	−10,536284
f_{14}	68 399	134 143	150 000	9,9978E-5 (8,5678E-5)	1,8690E-4 (2,6150E-5)	1,6E-4 (7,3E-5)	0



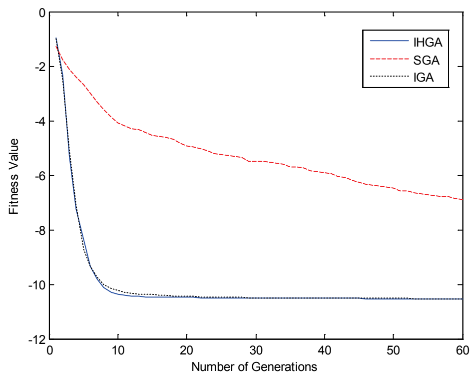
(a) Benchmark Function f_2



(b) Benchmark Function f_4



(c) Benchmark Function f_7



(d) Benchmark Function f_{13}

Figure 3 Average evolution comparison curve of three algorithms on the test functions running 20 times

According to the above test function, this paper adopts the improved hybrid genetic algorithm (IHGA) and conventional genetic algorithm (SGA) and the improved real-coded genetic algorithm (IGA) test on 14 benchmark function, and compare the performance of different algorithms. In the comparative experiments, the algorithm of parameter selection in accordance with the following: three different algorithms of population size P is P = 30; SGA with fitness proportional selection, arithmetic crossover and uniform mutation operator, the crossover probability p_c = 0,8, the mutation probability p_m = 0,02; IGA algorithm with nonlinear sorting options, environmental evolution crossover operator and non-uniform mutation operator, including the choice of the best individual probability q = 0,1, crossover probability p_c = 0,8, the mutation probability p_m = 0,02, environmental evolution in the crossover operator $\omega_{1-4} \in [0,1]$; total number of the next generation of M = 0,1P produced by the particle swarm optimization in this hybrid algorithm, the compression factor χ = 0,729, particle velocity updating coefficients ϕ_1 = 2,8, ϕ_2 = 1,3. When the overall algorithm evolving each generation, the particle swarm optimization strategy of individual PSO updates 10 times, selection, crossover and mutation probability in the genetic operation parameters as IGA algorithm choice. We adopt the three above different algorithms on 14 benchmark test functions independently run 20 times and record the following results: (1) the average fitness value; (2) the fitness value of standard deviation. Each test function of three algorithms according to different function of complexity takes different times of iterations, as shown in Tab. 2.

You can see the simulation results by comparison from Tab. 3:

(1) IHGA compared with SGA algorithm's optimization results, according to the function of $f_1 - f_4, f_7$ and f_9, IHGA algorithm has achieved the global minimum value of zero and the variance is zero, this shows that IHGA algorithm significantly improves the global search ability of the algorithm, and has good robustness, other test function of the result also shows that optimization ability of the hybrid algorithm is better than conventional genetic algorithm's; Compared with IGA algorithm optimization results, IHGA algorithm function gets the optimal solution of the test quality better than the IGA algorithm.

(2) From 20 times of independent operation to get quality stability in optimization solution, except IHGA f_{13} test function slightly more than the standard deviation of IGA, the fitness value of the IHGA algorithm to get the standard deviation is smaller than other two algorithms, instructions on solving quality IHGA algorithm has more stability.

For the image of the three algorithms extreme optimization evolutionary process, Fig. 3 depicts the average fitness value of the evolutionary curve generated by the three algorithms for some of the test functions running independently 20 times.

From the evolution curve in Fig. 3 can be seen, IHGA algorithm can fast convergence to the global optimal solution with relatively less evolution algebra, has strong global search ability and fast convergence speed. It effectively reduces the conventional genetic algorithm premature convergence which easily be trapped into local optimal solution phenomenon.

5.2 IHGA and FEP [15], OGA/Q [16] performance experiment

Comparative studies in recent years, the optimization of some existing algorithms of superior performance, from the optimization results can be obtained, OGA/Q, FEP algorithm to optimize the solution accuracy and robust algorithm has better performance than some other algorithms, the superiority of the hybrid genetic algorithm for the performance to verify improvement later. In order to increase the contrast, this article uses table 1 to list 14 benchmark functions to compare the improve optimization performance of hybrid genetic algorithm with FEP algorithm and OGA/Q.

In the computational experiments, the parameters selection of the hybrid algorithm are the same settings as before, for each test function evolution algebra T are shown in Tab. 2. When the current iteration number τ reaches the evolution algebra T , the algorithm stops.

During the test, because the hybrid algorithm uses the evolution of different parameters and termination conditions, this paper in the process of comparison, precision and quality advantages comparing their final algorithm optimization solution of the hybrid algorithm is less than the above two algorithms in computational case. So like FEP, OGA/Q algorithm, hybrid optimization runs independent test for each benchmark function extreme value 20 times and records the following results: (1) the fitness value function to calculate the number of times; (2) the average fitness value; (3) the fitness value of standard deviation.

From Tab. 4 can be found, to 14 functions f_1, f_2, f_3, f_4, f_7 and f_9 IHGA algorithm optimization can find the optimal value, and the number of evaluations is much less than FEP and OGA/Q algorithm. Compared with the FEP algorithm, both in solution quality and evaluation of the function of frequency, the hybrid algorithm is much better than the FEP algorithm, and has 6 functions of hybrid algorithm to find the optimal solution 0. Compared with the OGA/Q algorithm, the f_8 function, quality of mixed optimal solution algorithm is slightly larger than the OGA/Q algorithm, but the standard deviation of hybrid algorithm is 0 but the standard deviation of OGA/Q algorithm is 3,9890E-17. This illustrates the robustness of the hybrid algorithm is better than OGA/Q algorithm. And the evaluation number of hybrid algorithm is 6864, and the evaluate number of OGA/Q algorithm is 112421, amount of computation is far greater than the hybrid algorithm, for function f_8, f_{10}, the standard deviation of hybrid algorithm is slightly larger than OGA/Q algorithm, but the optimal solutions of better quality than OGA/Q algorithm, and the number of hybrid algorithm evaluation to the two function is 68440, 68379, the number of OGA/Q algorithm on the two function is respectively 302166, 134556, calculating quantity is much greater than the hybrid algorithm. On the function, the resulting solution quality and standard deviation of hybrid algorithm are better than the OGA/Q algorithm.

To make a long story short, the results show that a hybrid algorithm shows good performance, for unconstrained optimization problem. On the one hand, it can find high quality solution; it required a number of times less, less time complexity. On the other hand, the variance of the solution of the hybrid algorithm is very small, which shows that performance of the hybrid algorithm is stable. And it has strong robustness.

6 Conclusion

This paper presents a combination of genetic algorithm and particle swarm optimization, hybrid IHGA algorithm. The algorithm firstly generates initial population by chaotic sequence. In order to maintain the population diversity and not entirely dependent on the fitness value of the absolute value of the information, this article uses the non-linear ranking selection operator. According to ecology niche construction theory, it designs environmental evolution crossover operator. As there are many evolutionary directions early and later evolutionary conservative in the natural process, it designs non-uniform mutation operator. By improving the three operators of GA algorithm, improved algorithm to optimize performance. At the same time, elite strategy was introduced. For the high fitness value of particles, it improves PSO algorithm to improve their ability to adapt to the environment by iteratively repeated. Testing the performance of IHGA algorithm through several benchmark functions , the results show that the hybrid algorithm has good global searching ability, fast convergence speed. And the robustness of solutions quality and optimization is well.

Acknowledgements

It is supported by Youth project of National Natural Science fund (41071262), Key Laboratory of intelligent education and information engineering project in Heilongjiang province, Heilongjiang province key discipline of computer application technology (081203), Harbin Normal University Doctor Scientific Research Foundation.

7 References

[1] Xiangning, Lin; Shuohao, Ke; Zhengtian, Li. A Fault Diagnosis Method of Power Systems Based on Improved Objective Function and Genetic Algorithm-Tabu Search. // IEEE Transactions on Power Delivery. 25, 3(2010), pp. 1268-1274.
[2] Guang-Jian, Tian; Yong, Xia; Yanning, Zhang. Hybrid Genetic and Variational Expectation-Maximization Algorithm for Gaussian-Mixture-Model-Based Brain MR Image Segmentation. // IEEE Transactions on Information Technology in Biomedicine. 15, 3(2011), pp. 373-380.
[3] Youngjun, Ahn; Jiseong, Park; Cheol-Gyun, Lee et al. Novel Memetic Algorithm implemented With GA (Genetic Algorithm) and MADS (Mesh Adaptive Direct Search) for Optimal Design of Electromagnetic System. // IEEE Transactions on Magnetics. 46, 6(2010), pp. 1982-1985
[4] Aminifar, F.; Lucas, C.; Khodaei, A. et al. Optimal Placement of Phasor Measurement Units Using Immunity Genetic Algorithm. // IEEE Transactions on IPower Delivery. 24, 3(2009), pp. 1014-1020.
[5] Yi-Tung, Kao; Erwie, Zahara. A hybrid genetic algorithm and particle swarm optimization for multimodal functions. // Applied Soft Computing. 8, (2008), pp. 849-857.
[6] Wei, Cheng; Haoshan, Shi; Xipeng, Yin et al. An Elitism Strategy Based Genetic Algorithm for Streaming Pattern Discovery in Wireless Sensor Networks. // IEEE Communications Letters. 15, 4(2011), pp. 419-421.
[7] Gonzalez-Longatt, F. M.; Wall, P.; Regulski, P. et al. Optimal Electric Network Design for a Large Offshore Wind Farm Based on a Modified Genetic Algorithm Approach. // IEEE Systems Journal. 6, 1(2012), pp. 164-172.
[8] Holland, J. H. Adaptation in Natural and Artificial Systems. The University of Michigan Press,1975
[9] Han, Xiao-zhuo; Li, Zi-zhen; Hui, Cang et al. Theory of niche construction and application of its two-locus population genetic model. // Acta Botanica Boreali-Occidentalia Sinica. 24, 3(2004), pp. 558-562.
[10] Yan, X. F.; Chen, D. Z.; Hu, S. X. Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model. // Computers and Chemical Engineering. 27, 10(2003), pp. 1393-1404.
[11] Das Sharma, K.; Chatterjee, A.; Rakshit, A. A Random Spatial lbest PSO-Based Hybrid Strategy for Designing Adaptive Fuzzy Controllers for a Class of Nonlinear Systems. // IEEE Transactions on Instrumentation and Measurement. 61, 6(2012), pp. 1605-1621.
[12] Chia-Nan, Ko; Ying-Pin, Chang; Chia-Ju, Wu. A PSO Method with Nonlinear Time-Varying Evolution for Optimal Design of Harmonic Filters. // IEEE Transactions on Power Systems. 24, 1(2009), pp. 437-444.
[13] Vlachogiannis, J. G.; Lee, K. Y. Economic Load Dispatch – A Comparative Study on Heuristic Optimization Techniques with an Improved Coordinated Aggregation-Based PSO. // IEEE Transactions on Power Systems. 24, 2(2009), pp. 991-1001.
[14] Gao, J. Q.; He, G. X.; Wang, Y. Sh. Comparison and analysis with classic artificial immune algorithms on performance. // Computer Engineering and Applications. 45, 10(2009), pp. 208-210.
[15] Yao, X.; Liu, Y.; Lin, G. M. Evolutionary Programming Made Faster. // IEEE Transactions on Evolutionary Computation. 3, 2(1999), pp. 82-102.
[16] Leung, Y. W.; Wang, Y. P. An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization. // IEEE Transactions on Evolutionary Computation. 5, 1(2001), pp. 41-53.

Authors' addresses

Weidong Ji, Associate Professor
Department of Computer Science and Information Engineering, Harbin Normal University
Harbin 150025, P. R. China
E-mail: kingjwd@126.com

Jianhua Wang, Professor
Department of Computer Science and Information Engineering, Harbin Normal University
Harbin 150025, P. R. China
E-mail: wjh@vip.163.com

Jun Zhang, Professor
Department of Computer Science and Information Engineering, Harbin Normal University
Harbin 150025, P. R. China
E-mail: hsd_zj@163.com