A New Genetic Algorithm for Optimization

Ji Peirong Hu Xinyu Zhao Qing
College of Electrical engineering and information technology
China Three Gorges University
Yichang 443002, P.R.China
jipeirong@163.com

***Abstract*—A pseudo-parallel chaotic genetic algorithm is presented in this paper. The proposed algorithm is established by using the pseudo-random property of chaotic sequence and putting chaos into a pseudo-parallel genetic algorithm. The calculating results of three testing functions demonstrate that the both of the premature phenomenon and the slow convergence in conventional standard genetic algorithm can be prominently improved with the algorithm, and the presented algorithm is also superior to the pseudo-parallel genetic algorithms in the aspect of avoiding premature**.

***Keywords*—cgenetic algorithms; chaos optimization; pseudo-parallel genetic algorithms**

I. INTRODUCTION

During the last three decades, there has been a growing interest in algorithms which rely on analogies to natural phenomena such as evolution, heredity, and immunity. The standard genetic algorithm (SGA) [1-3] belongs to one category of these best known algorithms, and was proposed in 1975 by J. Holland, a professor of Michigan university of USA.

The SGA has many advantages and is characterized by the following components: a genetic representation (or an encoding) for the feasible solutions to the optimization problem, a population of encoded solutions, a fitness function that evaluates the optimality of each solution, control parameters. If certain conditions are satisfied, it is said to converge to the global optimum solution for studied systems. Moreover, it is easy to deal with integer problem, so it has been widely used in many fields .At the same time, disadvantages of SGA have appeared: 1) the calculating time is sometimes too long, 2) the optimal solution cannot be assured for premature problem mainly caused by lack of diversity, because all the genetic operators, such as selection, crossover, and mutation, are imposed on the whole population, most of the individuals might have tendency towards the same. In order to overcome the two shortcomings of SGA, many kinds of modified genetic algorithms have been proposed, one of the modified genetic algorithms is the pseudo-parallel genetic algorithm (PPGA) [4]. It can obviously improve the premature phenomenon and the slow convergence problem, and has been used in some fields [5,6] . Based on the PPGA, the pseudo-parallel chaotic genetic algorithm (PPCGA) is proposed by putting chaos into the PPGA in this paper.

II. EXCHANGE MODEL OF EVOLUTION INFORMATION

The pseudo-parallel genetic algorithm (PPGA) is difference from the distributed parallel genetic algorithm [4] by not using parallel computers but executing serially in a single computer, but the exchange model of evolution information in the algorithm is the same as the distributed parallel genetic algorithm. It is said that the distributed parallel genetic algorithm has advantages of premature-avoidance and linear acceleration. In the algorithm the whole population is divided into many sub-populations evolving independently, and each sub-population can exchange evolution information with other sub-populations at appropriate time. In this way, some excellent individuals can be exchanged to increase the diversity of population, and then the better solution will be got. There are three models [4] of exchange of evolution information, e.g. island model, stepping-stone model, neighborhood model.

III. PSEUDO-PARALLEL CHAOTIC GENETIC ALGORITHMS

The PPGA can simulate the different relationships between nature lives during the evolution process. Because of the difference in evolution manners of subpopulations, algorithm performance can be affected obviously. The evolution theory [7] has indicated that the evolution process is not of constant speed for nature lives. According to the theory, by introducing chaos into the evolution process of subpopulations of PPGA, we propose the pseudo-parallel chaotic genetic algorithm.

Chaos is a normal nonlinear phenomenon in nature. It is not disordered, and has some special dynamic characteristics: sensitive to initial value, pseudo random and universal [8] .The universal characteristic of chaos systems means that the chaotic movement can non-repeatedly have all states in certain range according to some rules. Therefore introducing chaos into each stage of PPGA can make the algorithm much closer to the real evolution process, better results will be got.

We mainly use following mappings

$$x_{k+1} = f(x_k) = \mu x_k (1 - x_k) \qquad x_k \in (0,1) \tag{1}$$

$$x_{k+1} = f(x_k) = \begin{cases} 2x_k & x_k \in [0, 0.5) \\ 2(1 - x_k) & x_k \in [0.5, 1] \end{cases} \tag{2}$$

For equation (1), when μ=4, the system is completely in chaotic status.

A. Introduce Chaos into Production of Initial Population

In SGA, the initial population is produced randomly. In this way, the individuals of initial population might gather up, so the loss of diversity of population occurs. In order to make the individuals of initial population be uniformly distributed in the whole solution space, we should take advantage of the universal characteristic of chaos to produce initial population.

Let the size of population be M , there are P variables in the unsolved problem. Step 1: By using equation (1) and initializing x_0 at any value ($x_0 \neq 0.25, 0.5, 0.75$), P chaos loci can be got. Step 2: Extract M items which are be gaining at the 30th item and ending at the 29+M th item. Step 3: Make pairs according to the order of the extracted M items. Step 4: Map them to the feasible field of unsolved problem.

B. Introduce Chaos into Mutation Operator Units

Following equations are used

$$y_i'(j_l) = y_i(j_l) + \alpha\beta x_i(j_l) \quad (3)$$

$$\alpha = 1 - (\frac{i}{i+k})^c \quad (4)$$

$$\beta = (-1)^b \quad (5)$$

$$i = 1,2,\cdots,G; j = 1,2,\cdots,M; l = 1,2,\cdots,p .$$

Where, G is the maximum number of evolution generations, M is the size of population, and p is the number of variables of the unsolved problems. $y_i(j_l)$ is the value which comes from mapping the $l-th$ element of individuals to chaos area (0,1), $y_i'(j_l)$ which is needed to map back to the code area and valued between (0,1) is the value of the $l-th$ element of individual after mutation, and $x_i(j_l)$ produced by equation (2) is the chaos variable which is added to the $l-th$ element of individuals. c , k are two integers with the ranges of $c > 1$ and $k > 1$, they are used to adjust the number of mutation. b is a random integer, which can make the mutation be randomly done in the directions of increasing or decreasing.

Form equation (3) we can know that the mutation operator simulates the process of evolution in nature, so the global optimal solution can be got.

C. The Process of Pseudo-parallel Chaotic Genetic Algorithm

The general form of optimization problem is :

$$\max \ f(x)$$

$$s.t \ \ x_i \in [a_i, b_i] \ \ i = 1,2,\cdots,n \ \ x = (x_1, x_2, \cdots, x_n)$$

The basic steps of the pseudo-parallel chaotic genetic algorithm are:

(1). Initialize and determine the parameters.

Size of population is M , maximum value of evolution generation is G , crossover probability is p_c and mutation probability is p_m . Follow the method showed in 3.1 to produce initial population, divide population $p(t)$ into subpopulation by using the selected exchange model of evolution information:

$$p(t) = \{p_1(t), p_2(t), \cdots, p_n(t)\}$$

Where n is the number of group, t is the number of generation.

(2).Compute the fitness value of each individual

(3).Do evolution to each group $p_i(t)$.

a. Reproduce .Use usual operator, such as roulette wheel selection etc..

b. Crossover to current subpopulation. According to the crossover probability, randomly select two individuals as fathers from reproduced subpopulation.

c. Mutation. Mutate individuals of current subpopulation, according to the method showed in 3.2.

(4). Compute fitness value of each individual according to different groups.

(5). Exchange information among $p_i(t)$ by a kind of exchange model of evolution information, get the population of the next generation.

(6). Output results and stop algorithm if the termination condition is met; or turn to (3) if don't.

IV. SIMULATIONS

In order to evaluate the performance of the proposed algorithm, following three typical test functions will be used.

F1: $$\max f(x_1, x_2) = 100(x_1 - x_2)^2 + (1 - x_1)^2$$
$$x_i \in [-2.048, 2.048]$$

F2: $$\min f(x, y) = [1 + (x + y + 1)^2 (19 - 14x + 3x^2 - 14y + 6xy + 3y^2)] \times [30 + (2x - 3y)^2 (18 - 32x + 12x^2 + 48y - 36xy + 27y^2)]$$
$$x, y \in [-2, 2];$$

F3: $$\min \ f(x, y) = \{\sum_{i=1}^{5} i\cos[(i+1)x+1]\}\{\sum_{i=1}^{5} i\cos[(i+1)y+1]\} + 0.5[(x + 1.42513)^2 + (y + 0.80032)^2]$$
$$x, y \in (-10, 10)$$

SGA, PPGA and PPCGA are been used to solve above 3 test functions. When testing, the population size $M = 120$, maximum evolution times $G = 30$,crossover probability $p_c = 0.6$,mutation probability $p_m = 0.04$. In PPGA and PPCGA, the population is divided into 4 groups of subpopulation and each has 30 individuals. The exchange model of evolution information is stepping-stone model.

Under the specific parameter condition, 200 tests are finished in accordance with each test function and each algorithm. The results are shown in table 1, where N_0 means the times of convergence to global optimal solution, N_1 means the times of premature phenomenon , R means success ratio, N_2 means the average evolution times of getting global optimal solution.

Table 1 The test results of SGA,CGA,PPCGA（200times）

	Algorithms	N_0	N_1	R	N_2
F1	SGA	64	136	32%	58
	PPGA	139	61	69.5%	38
	PPCGA	176	24	88%	30
F2	SGA	110	90	55%	158
	PPGA	164	36	82%	109
	PPCGA	187	13	93.5%	101
F3	SGA	134	66	67%	97
	PPGA	167	33	83.5%	42
	PPCGA	183	17	91.5%	38

From the Table 1, we can know PPCGA is a better algorithm than SGA in the aspects of avoiding premature and convergence, and superior to PPGA in the aspect of avoiding premature.

V. CONCLUSIONS

A pseudo-parallel chaotic genetic algorithm for optimization is presented. Three testing functions are used to verify the effectiveness of the proposed algorithm, simulation results show that PPCGA is superior to SGA in the aspects of avoiding premature and convergence, and superior to PPGA in the aspect of avoiding premature. The good performance of the proposed algorithm indicates that pseudo-parallel chaotic genetic algorithm presented in this paper is a promising and feasible method.

REFERENCES

[1] [1] Holland JH. Adaptation in Natural and Artificial systems[M]. Ann Arbor: The University of Michigan Press, 1975.

[2] [2] Li Minqiang, Kou Jisong, Lin Dan, etc. Genetic Algorithm: Basic Theory and Application [M]. Beijing: Science Press, 2002.

[3] [3] Wang Xiaoping, Cao Liming. Genetic algorithms: Theory, application and software realization [M]. Xi'an: Xi'an Jiao tong University Press, 2002.

[4] [4] Zhou Ming, Sun Shudong. Genetic Algorithm: Theory and Applications [M]. Beijing: National Defense Industry Press, 1999.

[5] Yang Jingtao, Zou Jianzhang, Xiang Ling, etc. Pseudo parallel genetic algorithm and its application in economic operation of hydropower station [J]. Power System Technology, 2004, 28(14): 53-56

[6] Wang, Z.H. ;Yin, X.G.; Zhang, Z.; etc. Pseudo-Parallel Genetic Algorithm for Reactive Power Optimization [M], 2003 IEEE Power Engineering Society General Meeting Conference Proceedings, v 2, 2003: 903-907

[7] Zhang Yun. Rate of Evolution and Unification of Evolution Theories[J]. Journal of Peking University, 1997, 33(6):754-803

[8] Wu X X. Chen Z. Introduction of Chaos Theory [M], Shanghai: Shanghai Science and Technology Bibliographic Publishing House, 1996