

# Hybridizing Adaptive Genetic Algorithm with Chaos Searching Technique for Numerical Optimization

Dongping Tian<sup>1,2</sup>

<sup>1</sup> *Institute of Computer Software, Baoji University of Arts and Sciences,  
Baoji, Shaanxi, 721007, China*

<sup>2</sup> *Institute of Computational Information Science,  
Baoji University of Arts and Sciences, Baoji, Shaanxi, 721007, China  
{tdp211}@163.com*

## Abstract

*Genetic algorithm (GA) is a population-based approach for heuristic search in optimization problems based on the principle of biologic evolution and natural selection. In this paper, we present a hybrid adaptive genetic algorithm with chaos searching technique for numerical optimization. On the one hand, two sets of crossover and mutation rates are formulated to automatically maintain the balance between exploration and exploitation during the genetic search process. On the other hand, the chaos searching technique is introduced into the adaptive genetic algorithm based on the decision mechanism for premature convergence adopted in this paper, whose main goal is to avoid being trapped into the local optimum. In addition, half of the total evolutionary generation is utilized as one of the decision conditions so as to speed up the convergent process. To validate the effectiveness and efficiency of the proposed approach, we apply it to four benchmark functions obtained from the literature, and the experimental results show that the proposed algorithm can find global optimal or the closer-to-optimal solutions and have faster search speed as well as higher convergence rate.*

**Keywords:** *Adaptive genetic algorithm (AGA), Chaos searching, Exploration, Exploitation, Hybrid soft computing (HSC)*

## 1. Introduction

Genetic algorithm [1] belongs to a category of stochastic search techniques. Ever since its introduction by professor Holland in Michigan University of America in 1970s, GA has been widely applied to solve linear and nonlinear problems by mimicking the evolutionary process from natural genetics and until now, it has become one of the significant intelligent computations in the 21st century. As we know, GA provides a common framework for solving optimization problems in complex systems, and it does not depend on the domain-specific of candidate problems. So as far as the problem classification is concerned, GA owns strong robustness [2]. However, in standard genetic algorithm (SGA), due to the rates of crossover and mutation operators are consistently fixed during the whole genetic search process, so it is not necessarily efficient when applied to solve complex multi-variable problems, especially in engineering applications. Furthermore, there inevitably exists premature convergence. It has been proved that although SGA can converge to 90 percent of the optimal solutions at a fast speed, it indeed will take much longer time to reach the real optimal solutions [3]. To address these issues, many researchers have made great efforts to improve the performance of GA. It's worth noting that the significance of the probabilities of crossover and mutation in controlling GA's performance has long been acknowledged, since whether the algorithm will find a near optimum solution or whether it will find a

solution efficiently [4] is heavily dependent on them. Instead of using fixed crossover and mutation rates, some adaptive parameter control schemes that can relieve the burden of specifying them have been proposed [5], *i.e.*, adaptive genetic algorithm (AGA), in which crossover and mutation rates varied adaptively depending on the fitness values of the solutions. However, even if AGA can accelerate its convergent speed and excel the standard GA in astringency and global searching ability, it also has higher possibility to get stuck at the local optimum because crossover and mutation rates of the best solutions with largest fitness values will both equal zero, which means that the crossover and mutation will not occur any more for the best solutions.

Facing with the above problems, a significant amount of work has been devoted to improving the existing AGA, especially the crossover and mutation rates, so as to make it get better balance between exploration and exploitation as well as fast convergence and global optimum to solve the practical problems effectively. Related results indicate that the performance of AGA can be significantly improved by introducing chaos optimization strategy and simulating some biological evolution elements and mechanisms, such as sexual reproduction, Baldwin effect and adaptation. Early work by Gao *et al.*[6] proposes a hybrid adaptive simulated annealing and genetic algorithm based on mutative scale chaos optimization strategy to solve the slow convergence and incident getting into local optimum characteristics of the standard GA, which combines the parallel searching structure of GA with the probabilistic jumping property of simulated annealing. Chen *et al* [7] put forward a novel chaos genetic algorithm combining chaotic search with self-adaptive GA for globally optimizing the initial weight and threshold of neural network-based state model for pipe network, which mainly uses the characteristics of chaos, *i.e.*, sensitive dependence on initial conditions, intrinsic stochastic property and ergodicity of chaotic motion. Followed by Chen *et al.*[8] present a chaos-genetic hybrid algorithm with a new random number generator using the mixed congruential method. Recently, Zhang *et al.*[9] propose a hybrid self-adaptive GA that utilizes a gender determination method to determine the gender of individuals in population. Then it adopts the different initial genetic parameters for female and male subgroups, and self-adaptively adjusts the sexual genetic operation based on the competition and cooperation between different gender subgroups. Zhuang *et al.*[10] employ a good-point set theory to produce a uniform initial population so as to enhance the optimization efficiency of the genetic algorithm. At the same time, the crossover and mutation probabilities are improved by using the sigmoid function and they can be adjusted nonlinearly between average fitness and maximal fitness with individual fitness. In a more recent work by Gao *et al.*[11], a rapid GA based on chaos mechanism is proposed, whose aim also is to use the ergodic property of the chaos movement so as to remedy the defect of premature convergence in the genetic algorithm. In addition, a cloud-model-based adaptive GA is presented by Dai *et al.*[12], which presents the use of cloud model to adaptively tune the probabilities of crossover and mutation depending on the fitness values of solutions in order to realize the twin goals of maintaining diversity in the population and sustaining the convergence capacity of the GA. Meanwhile, the other improved versions of AGA have also been proposed in most recent years [13, 14].

As briefly reviewed above, most of these approaches can achieve ideal optimization performance and motivate us to explore better AGA models with the help of their excellent experiences and knowledge. So in this paper, a new hybrid soft computing is proposed, which integrates AGA with chaos searching technique based on two sets of crossover and mutation rates to automatically maintain the balance between exploration and exploitation as well as the chaos searching into AGA to avoid being trapped into the local optimum. The rest of the paper is organized as follows. Section 2 provides a comprehensive survey on the strategies of dynamic adjustments for crossover and mutation rates by AGA. In section 3, the strategies of dynamic adjustments of crossover and mutation rates with heuristics are discussed. Section 4 elaborates the hybrid AGA

with chaos searching technique proposed in this paper. Numerical simulation results and their comparisons are reported in section 5. Finally, we end this paper with some important conclusions and future research directions in section 6.

## 2. Dynamic Adjustments of Crossover and Mutation Rates by AGA Itself

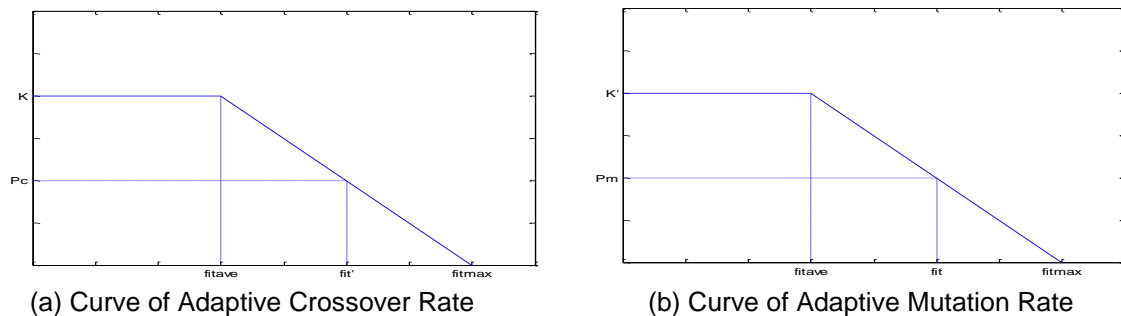
### 2.1 Dynamic Linear Adjustments

**2.1.1 Standard Adaptive Genetic Algorithm (SAGA):** The SAGA is proposed by Srinivas in 1994 [5]. Its main idea is that when the fitness values of population tend to convergence, the probability of the occurrence of the genetic operators will be increased so as to avoid the premature convergence, whereas when the fitness values of population tend to divergence, the probability of the occurrence of genetic operators will be decreased so as to converge to the optimum. The probabilities of crossover and mutation are defined as follows:

$$p_c = \begin{cases} k_1 \frac{fit_{max} - fit'}{fit_{max} - fit_{ave}}, & fit' \geq fit_{ave} \\ k_3, & fit' < fit_{ave} \end{cases} \quad (1)$$

$$p_m = \begin{cases} k_2 \frac{fit_{max} - fit}{fit_{max} - fit_{ave}}, & fit \geq fit_{ave} \\ k_4, & fit < fit_{ave} \end{cases} \quad (2)$$

where  $fit_{ave}$  and  $fit_{max}$  denote the average and maximal fitness values of the population respectively,  $fit'$  is the larger fitness value of the two individuals to be crossed, and  $fit$  is the fitness value of the individual to be mutated.  $k_1$ ,  $k_2$ ,  $k_3$  and  $k_4$  are constants predetermined in the range [0,1]. From Eqs.(1) and (2), it is easy to see that the linear mapping relations between the fitness values of population and the crossover and mutation rates shown in Figure 1, where  $k=k_1=k_3$ , and  $k'=k_2=k_4$ .



**Figure 1. Dynamic Curves of Adaptive Crossover and Mutation Rates in SAGA**

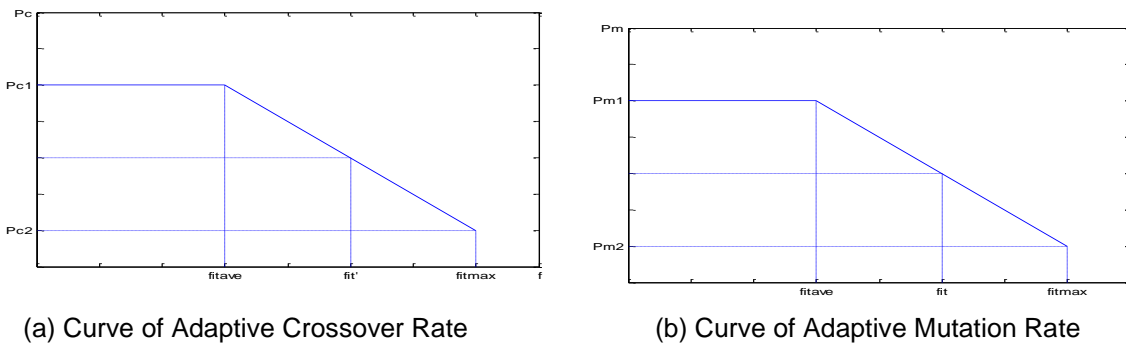
**2.1.2 Improved Adaptive Genetic Algorithm (IAGA):** (1) IAGA-1. In AGA, when the fitness value of individual is less than the average fitness value of the population, it indicates that the performance is poor and the crossover and mutation rates should be increased. When the fitness value of individual is greater than the average fitness value of the population, it implies that the performance is good and the corresponding crossover and mutation rates should be adopted. Obviously, when the fitness value tends to the maximum, the crossover and mutation rates in general become smaller progressively. And when the fitness value is equal to the maximum, the crossover and mutation rates will be zero. This strategy of adjusting crossover and mutation rates is suitable for the later

evolution of the population, but for the former stage of evolution, in actual fact, it will slow down the genetic search process. The reason lies in that the better individuals in the former stage almost unchanged, and at the time the best individual is not necessarily the global optimum of the candidate problem, which increases the probability of the occurrence for premature convergence. Hence, it is necessary to improve SAGA and just as described in [15]. The definitions of crossover and mutation rates are shown as follows:

$$p_c = \begin{cases} p_{c1} - \frac{p_{c1} - p_{c2}}{fit_{max} - fit_{ave}} * (fit' - fit_{ave}), & fit' \geq fit_{ave} \\ p_{c1}, & fit' < fit_{ave} \end{cases} \quad (3)$$

$$p_m = \begin{cases} p_{m1} - \frac{p_{m1} - p_{m2}}{fit_{max} - fit_{ave}} * (fit_{max} - fit), & fit \geq fit_{ave} \\ p_{m1}, & fit < fit_{ave} \end{cases} \quad (4)$$

Note that the meaning of each variable is the same as those described above except for  $p_{c1}=0.9$ ,  $p_{c2}=0.6$ ,  $p_{m1}=0.1$  and  $p_{m2}=0.001$ . Figure 2 graphs the linear mapping relations corresponding to Eqs.(3) and (4), respectively.



(a) Curve of Adaptive Crossover Rate (b) Curve of Adaptive Mutation Rate  
**Figure 2. Dynamic Curves of Adaptive Crossover and Mutation Rates in IAGA-1**

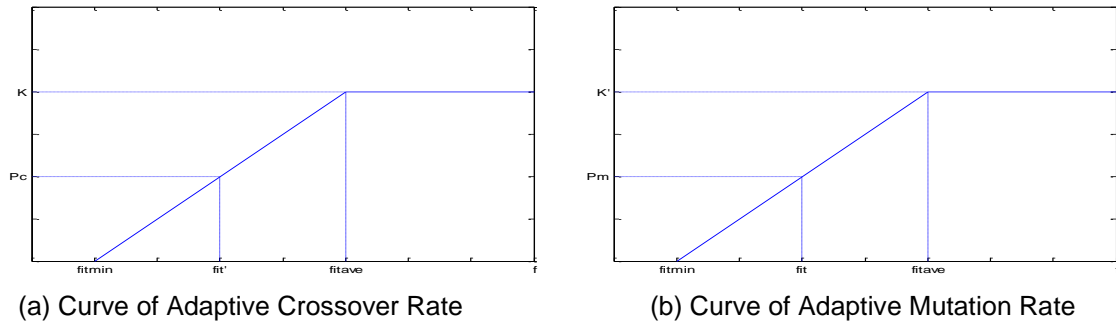
(2) IAGA-2

Due to the standard adaptive genetic algorithm has a tendency to stagnate in the former stage of evolution and easily get stuck at the local optimum in the latter, so an improved AGA has been introduced in Ref.[16], in which the crossover and mutation rates are defined as follows:

$$p_c = \begin{cases} k_1 \frac{fit' - fit_{min}}{fit_{ave} - fit_{min}}, & fit' < fit_{ave} \\ k_3, & fit' \geq fit_{ave} \end{cases} \quad (5)$$

$$p_m = \begin{cases} k_2 \frac{fit - fit_{min}}{fit_{ave} - fit_{min}}, & fit < fit_{ave} \\ k_4, & fit \geq fit_{ave} \end{cases} \quad (6)$$

Without loss of generality, the meaning of each variable is just the same as those described above and  $k=k_1=k_3$ ,  $k'=k_2=k_4$ . Figure 3 illustrates its linear mapping relations corresponding to the Eqs.(5) and (6), respectively.



**Figure 3. Dynamic Curves of Adaptive Crossover and Mutation Rates in IAGA-2**

In IAGA2, the premature convergence can be effectively prevented by increasing the cross-over and mutation rates of better individuals with Eqs.(5) and (6). In this way, however, it will inevitably slow down the evolutionary process and convergence speed.

### (3) IAGA-3

Ref.[17], according to the centralized degrees of fitness values, adaptively adjusts the crossover and mutation rates by adopting the maximal fitness value ( $fit_{max}$ ), minimal fitness value ( $fit_{min}$ ) and average fitness value ( $fit_{ave}$ ) of the population. The similarity of  $fit_{min}$  and  $fit_{max}$  reflects the centralized degrees of the whole population. If the value nears to 1, then GA, to a large extent, plunges into the local optimum, that is, the fitness values of population pool greatly. On the other hand, the similarity of  $fit_{ave}$  and  $fit_{max}$  reflects the distribution of fitness values internally. If it nears to 1, then all the individuals in the corresponding generation concentrate greatly. When  $(fit_{min}/fit_{max}) > b$  and  $(fit_{ave}/fit_{max}) > a$  (where  $a$  and  $b$  are positive real numbers in the range  $(0,1)$ ), the population can be viewed as “convergence” and at the time, the crossover and mutation rates adaptively adjust according to the population’s centralized degrees. Otherwise, it can be considered as “divergence” and the crossover and mutation rates should still maintain their initial values. The corresponding definitions are given as below:

$$p_c = \begin{cases} p_c * \frac{fit_{max}}{fit_{max} - fit_{min}}, \frac{fit_{ave}}{fit_{max}} > a \text{ (and)} \frac{fit_{min}}{fit_{max}} > b \\ p_c, \text{others.} \end{cases} \quad (7)$$

$$p_m = \begin{cases} p_m * \frac{fit_{max}}{fit_{max} - fit_{min}}, \frac{fit_{ave}}{fit_{max}} > a \text{ (and)} \frac{fit_{min}}{fit_{max}} > b \\ p_m, \text{others.} \end{cases} \quad (8)$$

Similarly, the meaning of each variable is the same as those described above except for  $p_c$  and  $p_m$ , which represent the initial rates of crossover and mutation operators, and  $0.5 < a < 1$ ,  $0 < b < 1$ .

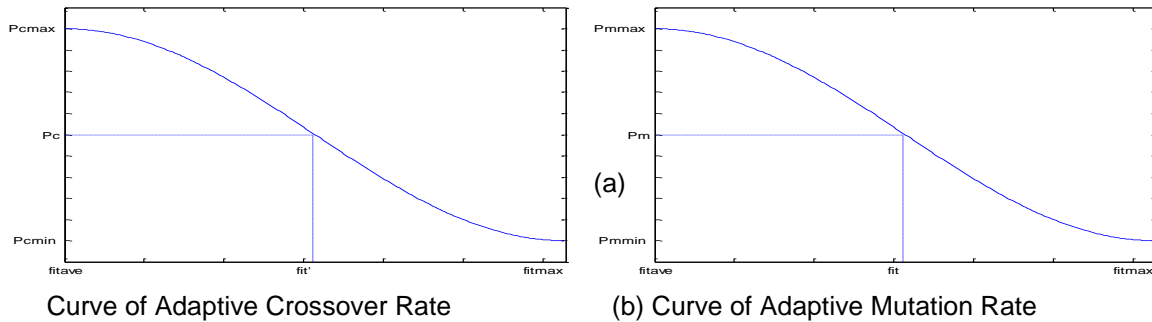
## 2.2 Dynamic Nonlinear Adjustments

**2.2.1 Improved AGA with Cosine Function (CAGA):** Due to the crossover and mutation rates have a significant effect on the performance of GA, so it is very important to choose the proper values of them during the genetic search process. Just as shown in Ref.[18], an improved AGA has been proposed by formulating the crossover and mutation operators as follows:

$$p_c = \begin{cases} \frac{p_{c\max} + p_{c\min}}{2} + \frac{p_{c\max} - p_{c\min}}{2} \cos\left(\frac{fit' - fit_{ave}}{fit_{\max} - fit_{ave}} \pi\right), & fit' \geq fit_{ave} \\ p_{c\max}, & fit' < fit_{ave} \end{cases} \quad (9)$$

$$p_m = \begin{cases} \frac{p_{m\max} + p_{m\min}}{2} + \frac{p_{m\max} - p_{m\min}}{2} \cos\left(\frac{fit - fit_{ave}}{fit_{\max} - fit_{ave}} \pi\right), & fit \geq fit_{ave} \\ p_{m\max}, & fit < fit_{ave} \end{cases} \quad (10)$$

where  $p_{c\max}$  and  $p_{c\min}$  represent the maximal and the minimal crossover rates, respectively. Similarly,  $p_{m\max}$  and  $p_{m\min}$  denote the maximal and the minimal mutation rates. The meaning of the rest variables is the same as those described above. Relatively, the adaptive curves are visualized in Figure 4.



**Figure 4. Dynamic Curves of Adaptive Crossover and Mutation Rates in CAGA**

As can be seen from Figure 4, CAGA promotes the crossover and mutation rates of individuals corresponding to the fitness values in the interval  $[fit_{ave}, (fit_{ave} + fit_{\max})/2]$ , whereas degrades them in the interval  $[(fit_{ave} + fit_{\max})/2, fit_{\max}]$ . Obviously, the purpose of doing this is to generate more new schemas, and at the same time, reserve the better ones in the population so as to get a better balance between exploration and exploitation.

### 2.2.2 Improved Adaptive Genetic Algorithm (IAGA)

#### (1) INAGA-1

In Ref. [19], an improved AGA has been proposed, which can get an effective trade-off between convergence speed and global optimum. It should be noted that the devised cross-over and mutation rates are associated with the evolutionary generations and the fitness values of individuals, respectively. Their definitions are shown as below:

$$p_c(t) = \begin{cases} m_{imp}, m_{imp} > p_{c,\min} \\ p_{c,\min}, others. \end{cases} \quad (11)$$

$$p_m(t) = \begin{cases} m_{imp}, m_{imp} > p_{m,\min} \\ p_{m,\min}, others. \end{cases} \quad (12)$$

$$m_{imp} = \exp\left(-\left|\frac{f_{\max} - f(X_i)}{f_{\max}}\right|\right) \times \frac{1}{1 + t/T_{Gen}} \times p_{m,\max} \quad (13)$$

In Equation (11),  $m_{imp} = p_{c,\max} \times 2^{(-t/T_{Gen})}$  is an intermediate computing quantity.  $T_{Gen}$  is the maximal evolutionary generation predetermined, and  $t$  represents the current generation and is restricted to the interval  $[0, T_{Gen}]$ .  $p_{c,\max}$  and  $p_{c,\min}$  are the maximal and minimal crossover rates predetermined.  $p_c(t)$  denotes the current crossover rate. Especially in Equation (12), the  $m_{imp}$  is also an intermediate computing quantity described by Eq.(13).

$fit_{max}$  is the maximal fitness value in the current population,  $p_{m,max}$  and  $p_{m,min}$  are the maximal and minimal mutation rates pre-determined, respectively.  $f(X_i)$  represents the fitness value of the individual to be mutated,  $p_m(t)$  stands for the mutation rate of  $X_i$  in the  $t$ -th generation. For more details on them please refer to the corresponding literature.

(2) INAGA-2

In order to enhance the performance of AGA, a new adaptive mutation rate is devised in [20], which is closely related to the evolutionary generation and the fitness value of individual. To be specific, its description can be found as follows:

$$p_m(t) = \begin{cases} m_{imp}, m_{imp} > p_{m,min} \\ p_{m,min}, others. \end{cases} \quad (14)$$

$$m_{imp} = \begin{cases} \exp\left(-\left|\frac{f_{max} - f(X_i)}{f_{max}}\right|\right) \times \frac{1}{1+t/T_{Gen}} \times p_{m,max}, f_{max} > 0 \\ \exp\left(-\left|\frac{f_{max} - f(X_i)}{f_{ave}}\right|\right) \times \frac{1}{1+t/T_{Gen}} \times p_{m,max}, f_{max} \leq 0 \end{cases} \quad (15)$$

Note that the crossover rate is not mentioned here because of its definition is the same as that in Ref. 19] in subsection 2.2.2 (1).

**2.2.3 Improved AGA with Sigmoid Function (SAGA):** In Ref.[21], the crossover and mutation rates have been devised with the neural active function sigmoid so as to make the genetic search process proceed until the sought solutions are reached. The crossover and mutation rates, here, are defined as follows:

$$p_c = \begin{cases} \frac{p_{c,max} - p_{c,min}}{1 + \exp\left(A\left(\frac{2(fit' - fit_{ave})}{fit_{max} - fit_{ave}} - 1\right)\right)} + p_{c,min}, fit' \geq fit_{ave} \\ p_{c,min}, fit' < fit_{ave} \end{cases} \quad (16)$$

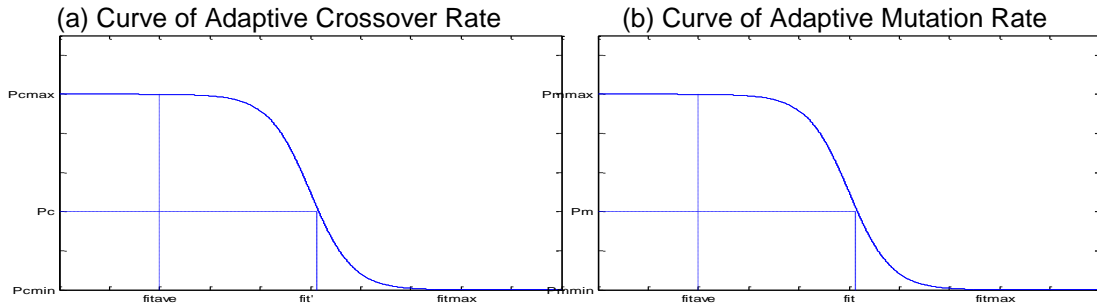
$$p_m = \begin{cases} \frac{p_{m,max} - p_{m,min}}{1 + \exp\left(A\left(\frac{2(fit - fit_{ave})}{fit_{max} - fit_{ave}} - 1\right)\right)} + p_{m,min}, fit \geq fit_{ave} \\ p_{m,min}, fit < fit_{ave} \end{cases} \quad (17)$$

$$p_c = \begin{cases} \frac{p_{c,max} - p_{c,min}}{1 + \exp\left(A\left(1 - \frac{2(fit_{ave} - fit')}{fit_{ave} - fit_{min}}\right)\right)} + p_{c,min}, fit' \leq fit_{ave} \\ p_{c,min}, fit' > fit_{ave} \end{cases} \quad (18)$$

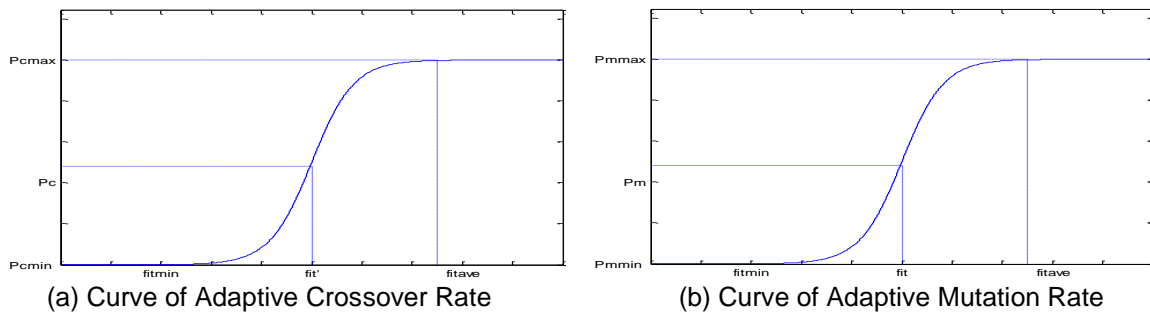
$$p_m = \begin{cases} \frac{p_{m,max} - p_{m,min}}{1 + \exp\left(A\left(1 - \frac{2(fit_{ave} - fit)}{fit_{ave} - fit_{min}}\right)\right)} + p_{m,min}, fit \leq fit_{ave} \\ p_{m,min}, fit > fit_{ave} \end{cases} \quad (19)$$

Similarly, the meaning of each variable is the same as those given above except for  $A$ , which is a constant predetermined with 9.903438. Among the two sets of formulas, (16) and (17) are the adaptive crossover and mutation rates for solving maximal optimization

problems, where- as (18) and (19) for the minimal ones. Their corresponding dynamic curves are illustrated in Figures 5 and 6, respectively.



**Figure 5. Dynamic Curves of Adaptive Crossover and Mutation Rates in SAGA (Max.)**



**Figure 6. Dynamic Curves of Adaptive Crossover and Mutation Rates in SAGA (Min.)**

### 3. Dynamic Adjustments of Crossover and Mutation Rates with Heuristics

#### 3.1 Heuristic 1

In this heuristic, the concept of the fitness values of parent and offspring at each generation is exploited by [22]. The scheme increases the occurrence rates of the crossover and mutation operators if it consistently produces a better offspring during genetic search process, however, it reduces the crossover and mutation rates, if it produces a poorer offspring. This scheme is based on the fact that it encourages the well-performing crossover and mutation operators to produce more offspring, while also reducing the chance for the poorly performing operators to destroy the potential individuals during the genetic search process. The detailed definition of the corresponding crossover and mutation rates is as follows:

$$p_c(t+1) = \begin{cases} p_c(t) + 0.05, & \overline{(f_{par\_size}(t))} / \overline{(f_{off\_size}(t))} - 1 \geq 0.1 \\ p_c(t) - 0.05, & \overline{(f_{par\_size}(t))} / \overline{(f_{off\_size}(t))} - 1 \leq -0.1 \\ p_c(t), & -0.1 < \overline{(f_{par\_size}(t))} / \overline{(f_{off\_size}(t))} - 1 < 0.1 \end{cases} \quad (20)$$

$$p_m(t+1) = \begin{cases} p_m(t) + 0.005, & \overline{(f_{par\_size}(t))} / \overline{(f_{off\_size}(t))} - 1 \geq 0.1 \\ p_m(t) - 0.005, & \overline{(f_{par\_size}(t))} / \overline{(f_{off\_size}(t))} - 1 \leq -0.1 \\ p_m(t), & -0.1 < \overline{(f_{par\_size}(t))} / \overline{(f_{off\_size}(t))} - 1 < 0.1 \end{cases} \quad (21)$$



Here,  $par\_size$  and  $off\_size$  are the parent size and offspring size satisfying constraints, respectively.  $f_{par\_size}(t)$  and  $f_{off\_size}(t)$  denote the average fitness values of parents and offspring at generation  $t$ , respectively.  $p_c(t)$  and  $p_m(t)$  are the crossover and mutation rates of the  $t$ -th generation.

### 3.2 Heuristic 2

In Ref.[23], an improved scheme has been proposed for reinforcing the mutation operator from the original work [5]. In the original work, it has been observed that the difference between the  $f_{max}$  and  $f_{ave}$  of the population likely to be less for a population that has converged to optimum solution than that for a population scattered in the solution space. Therefore, the crossover and mutation rates should be varied depending on the value of the  $f_{max}-f_{ave}$ . On the other hand, if the crossover and mutation rates have the same values for all the solutions in the population, which means that the solutions with high fitness values and the solutions with low fitness values are subjected to the same level of mutation and crossover. This will certainly deteriorate the performance of AGA. Hence, an improved mutation rate has been introduced by inserting an additional scheme into the original mutation scheme as below:

$$p_m = \begin{cases} C'(fit_i)k_2(fit_{max} - fit_i) / (fit_{max} - fit_{ave}), & fit_i > fit_{ave} \\ k_4, & fit_i \leq fit_{ave} \end{cases} \quad (22)$$

where

$$C'(fit_i) = \left( \sum_{j=1}^{pop\_size} |fit_i - fit_j| \right) / ((pop\_size - 1) \max_j |fit_i - fit_j|), i \neq j \quad (23)$$

Note that Equation (23) is an intermediate computing quantity,  $pop\_size$  denotes the population size.

### 3.3 Heuristic 3

In Ref.[24], a novel adaptive genetic algorithm has been proposed by employing the basic concept presented in literature [25] to adaptively regulate the AGA operators. The main scheme of this concept is to use two fuzzy logic controllers (FLC): the crossover FLC and mutation FLC are implemented independently to adaptively regulate the rates of crossover and mutation operators during the genetic search process. For minimal optimization problems, the change of the average fitness values at generation  $t$  can be calculated as follows:

$$\Delta fit_{ave}(t) = \overline{(f_{par\_size}(t) - f_{off\_size}(t))} \times \lambda \\ = \left( \frac{\sum_{k=1}^{par\_size} f_k(t)}{par\_size} - \frac{\sum_{k=par\_size+1}^{par\_size+off\_size} f_k(t)}{off\_size} \right) \times \lambda \quad (24)$$

where  $\lambda$  is a scaling factor to normalize average fitness for applying de-fuzzification in the FLC and is varied in accordance with the problem under consideration. The rates of crossover and mutation operators are calculated in the pseudo-code as below:

---

#### **Algorithm 1: Pseudo-Code for Crossover and Mutation Operators**

1. **Begin**
2. **if**  $\varepsilon \leq \Delta fit_{ave}(t-1) \leq \gamma$  and  $\varepsilon \leq \Delta fit_{ave}(t) \leq \gamma$ , **then**  
Increase  $p_c$  and  $p_m$  for next generation;
3. **if**  $-\gamma \leq \Delta fit_{ave}(t-1) \leq -\varepsilon$  and  $-\gamma \leq \Delta fit_{ave}(t) \leq -\varepsilon$ , **then**
4. Decrease  $p_c$  and  $p_m$  for next generation;

5. **if**  $-\varepsilon < \Delta fit_{ave}(t-1) < \varepsilon$  and  $-\varepsilon < \Delta fit_{ave}(t) < \varepsilon$ , **then**  
 Rapidly increase  $p_c$  and  $p_m$  for next generation.
6. **end if**
7. **End**

where  $\varepsilon$  is a given real number in the proximity of zero,  $\gamma$  and  $-\gamma$  are respectively a given maximum and minimum values of a fuzzy membership function.

## 4. Hybrid AGA with Chaos Searching Technique

Since 1990s, the GA, fuzzy theory, artificial neural networks, expert systems and chaotic optimization techniques have been widely applied in many fields. Furthermore, much attention has been paid to the integration of these approaches and thus the so-called hybrid soft computing (HSC) has become much popularized. GA, due to its robustness and better performance, has become very popular by researchers and played a crucial role in various HSCs. So in this paper, a new hybrid soft computing is presented by integrating AGA with chaos searching technique based on two sets of crossover and mutation rates to automatically maintain the balance between exploration and exploitation as well as the chaos searching into AGA to avoid being trapped into the local optimum. The details of it will be described in the following subsections.

### 4.1 Chaos Searching Technique

It is well known that chaotic map has the characteristics of uncertainty, ergodicity and stochastic properties, *etc.* Hence, the purpose of introducing chaotic search into AGA in this paper is to improve the exploration ability of the algorithm during the genetic search process. Generally, the basic idea of chaotic optimization is that [26], to begin with, mapping the chaotic variables in the chaotic space into the solution space by means of the logistic map, and then executing the search process repeatedly until the termination condition is met. The detailed procedure of chaotic optimization is devised as follows.

**Step 1:** Set  $m=0$ . Randomly generate  $D$  different locus chaotic variables  $cx_{dm}(d=1,2,\dots,D)$  except for the periodic fixed points (0,0.25,0.5,0.75,1), where  $d$  is the index of variable and  $m$  denotes the  $m$ -th chaos search.

**Step 2:** Transform  $cx_{dm}$  into the optimization variables  $rx_{dm}$  in the interval  $[a_d, b_d]$  by the following Eq.(3), where  $a_d$  and  $b_d$  are the lower and upper bounds of the optimization variables respectively.

$$rx_d^m = a_d^m + (b_d - a_d) * cx_d^m \quad (25)$$

**Step 3:** Execute chaos searching for  $x_{dm}$ :  $x_{dm} = x_{dm} + \beta \times rx_{dm}$ . If  $f(x_{dm}) < f^*$ , then set  $f^* = f(x_{dm})$  and  $x_{d^*} = x_{dm}$ , where  $f^*$  is the current optimal solution,  $x_{d^*}$  is the corresponding optimal variables and  $\beta$  is the step size.

**Step 4:**  $m=m+1$ ,  $cx_{dm} = 4 \times cx_{dm} \times (1 - cx_{dm})$ .

**Step 5:** Repeat steps 2, 3 and 4 until  $f^*$  keeps invariant or the maximal iteration steps is reached.

### 4.2 AGA with Chaos Searching Technique

The proposed approach is developed on the basis of the standard AGA via introducing chaotic search and the other set of crossover and mutation rates so as to guide the whole population to evolve in the solution space. In this method, the chaos searching algorithm is mainly utilized to obtain the local optimum for “local exploration” while the AGA with two sets of crossover and mutation rates is responsible for “global exploitation”. In order to fulfill the chaos searching, here, a decision identifier  $\varepsilon$  is employed for premature convergence [27]. Assume that  $f_{t,ave}$  is the average fitness value at generation  $t$ , which is computed by  $\sum_{i=1}^T f_{t,i} / T$ , where  $f_{t,i}$  represents the fitness value of the  $i$ -th individual in the

$t$ -th generation,  $T$  denotes the population size. At the same time, given that the best individual's fitness is  $f_{t,max}$ ,  $f_{ave}$  is the average of all the individuals whose fitness values are greater than  $f_{t,ave}$ . Then let  $\varepsilon = f_{t,max} - f_{ave}$ . It is well known that the rates of crossover and mutation operators have a significant effect on the convergence of AGA during the search process. Thus, the other set of crossover and mutation rates adopted in this paper is devised with  $\varepsilon$  as follows.

$$p_c = 1 + 0.4 \left( -\frac{1}{1 + \exp(-k_1 \varepsilon)} \right) \quad (26)$$

$$p_m = 0.04 + 0.04 \left( -\frac{1}{1 + \exp(-k_2 \varepsilon)} \right) \quad (27)$$

where  $k_1$  and  $k_2$  are the positive constants predetermined. Algorithm 1 illustrates the detailed procedure of the proposed algorithm in this paper.

---

### Algorithm 2: Pseudo-Code of the Proposed Algorithm

---

8. **Begin**
  9. Initialize population and various parameters,  $\varepsilon^*$ ,  $M$  and  $C$  denote the premature decision threshold, maximal number of iterations and current generation.
  10. **while**(maximal number of iterations is not met)
  11. Evaluate fitness and premature decision identifier  $\varepsilon$ .
  12. **if**  $\varepsilon \geq \varepsilon^*$  and  $C \geq 0.5 * M$
  13. **for**  $d=1$  to number of different initial chaotic variables
  14. Execute chaotic search as depicted in subsection 3.1.
  15. **next d**
  16. Output fitness value.
  17. **else if**  $\varepsilon \geq \varepsilon^*$  and  $C < 0.5 * M$
  18. Calculate crossover and mutation rates by Eqs.(25) and (26).
  19. Carry out the corresponding genetic operations.
  20. **else**
  21. Calculate crossover and mutation rates by the SAGA.
  22. Carry out the corresponding genetic operations.
  23. **end if**
  24. **end if**
  25. **next generation until stopping criterion**
  26. Output optimal solution.
  27. **End**
- 

## 5. Experimental Results and Analysis

To verify the performance of the proposed algorithm in this paper, four well-known bench- mark functions obtained from the literature are utilized here. Their expressions, search ranges and global optimum values are listed in Table 1.

**Table 1. Test Functions, Search Ranges and Global Optimum Values for the Test Functions**

Test function	Search range	Optimum val.
$f_1(x, y) = \left( 4 - 2.1x^2 + \frac{x^4}{3} \right) x^2 + xy + (-4 + 4y^2)y^2$	[-100,100]	-1.031628
$f_2(x, y) = \left\{ \sum_{i=1}^5 i \cos[(i+1)x + 1] \right\} * \left\{ \sum_{i=1}^5 i \cos[(i+1)y + 1] \right\}$	[-10,10]	-186.731

$f_3(x, y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{[1.0 + 0.001(x^2 + y^2)]^2}$	[-100,100]	1.0
$f_4(x, y) = 100(x^2 - y)^2 + (1 - x)^2$	[-2.048, -2.048]	0

It is noteworthy that function  $f_1$  has six local optimal solutions, among which (-0.0898, 0.7126) and (0.0898,-0.7126) are the global minimal points corresponding to the optimal solution -1.031628.  $f_2$  is a multimodal function with a considerable number of local optima in the region of interest, and its global optimal solution is -186.731 corresponding to eighteen points.  $f_3$  has only one global maximum 1.0 in its domain. The last simulation function  $f_4$  is a two-dimensional function and has one global minimum 0. Although it is a unimodal function, it is ill-conditioned and difficult to globally minimize. For fair comparison, the parameters for the proposed algorithm are set as follows: the population size for  $f_1$  and  $f_3$  is set to 150, while for  $f_2$  and  $f_4$  it is set to 40, the initial rates of crossover and mutation operators are set to 0.5 and 0.04, respectively.  $\beta$ , which is the step size for adjusting chaos search is set to 0.005. And the threshold value of the premature decision identifier  $\varepsilon^*$  is predetermined to be 0.1 by trial and error. In addition, fifty independent runs are executed for each different algorithm and each run loops 100 iterations.

For the sake of fair comparison, the best solution (BS), average solution (AS), minimal convergent generation (MCG) and average convergent generation (ACG) are utilized as metrics listed in Table 2. In addition, the sum of squared deviations (SSD) is also illustrated in Table 2 to further demonstrate the effectiveness of the proposed approach in this paper, which is defined as  $SSD = \sum_{i=1}^N (f_i - f_{ave})^2$ .

**Table 2. Performance Comparison of SGA, AGA and Ours**

Function	Method	AS	BS	ACG	MCG	SSD
$f_1$	SGA	-0.81616	-1.03162	15	36	1.7787e+001
	AGA	-0.86868	-1.03162	19	30	4.6976e+000
	Ours	-0.97097	-1.03163	26	27	4.2233e+000
$f_2$	SGA	-152.71239	-186.73073	16	35	1.1455e+006
	AGA	-170.71256	-186.73082	17	30	5.5424e+004
	Ours	-178.58870	-186.73094	24	20	5.4210e+004
$f_3$	SGA	0.93455	0.99032	5	49	6.5137e-001
	AGA	0.94519	0.99034	9	24	4.0786e-001
	Ours	0.95784	0.99412	9	44	2.1092e-001
$f_4$	SGA	1.25473	0.11283	6	48	2.6602e+003
	AGA	0.32995	0.08972	8	26	5.8573e+001
	Ours	0.13702	0.00028	12	11	1.8928e+001

From Table 2, it is easy to see that the performance of our approach significantly outperforms those of SGA and AGA. Taking the function  $f_1$  for example, the proposed algorithm leads to an increase of average convergent generation to 26 compared to 15 by SGA and 19 by AGA respectively. While for function  $f_3$ , it does not give rise to any increase, but its sum of squared deviation is obviously less than that of AGA, which indicates that the AGA incorporated with two sets of crossover and mutation rates together with the chaos searching can get a much better balance between exploration and exploitation in the solution space. In the meanwhile, it can effectively safeguard the genetic search process from being trapped into local optimal solutions. Similarly, other cases can also be discussed by the experimental results from Table 2. Alternatively, it

should be emphasized that in some cases the deviation of the results is rather small, but the difference in results is statistically very significant.

## 6. Conclusions and Future Work

This paper presents a novel hybrid soft computing method, which integrates AGA with chaos searching technique based on two sets of crossover and mutation rates to automatically maintain the balance between exploration and exploitation as well as the chaos searching into AGA to avoid being trapped into the local optimum. Experimental results on four well-known benchmark functions demonstrate the effectiveness and efficiency of the proposed algorithm, that is, it not only has the powerful ability to search the global optimum, but also can effectively prevent the premature convergence during the evolutionary process, at least for the four benchmark functions tested in this paper.

As future work, we plan to introduce this approach into some real-world research fields, such as integrated circuit design, media semantic understanding and engineering optimization scheduling, *etc.* Lastly, and arguably most importantly, the qualitative relationship between the crossover/mutation rates and the convergence of AGA, from the viewpoint of mathematics, will be elaborated and proved comprehensively.

## Acknowledgements

The work is supported by the National Natural Science Foundation of China (No.6137903-0), the National Program on Key Basic Research Project (No.2013CB329502), the Natural Science Basic Research Plan in Shaanxi Province of China (No.2014JQ2-6036) and the Science and Technology R&D Program of Baoji City (No. 203020013, No.2013R2-2).

## References

- [1] J. Holland, "Adaptation in natural and artificial systems", MIT Press, USA (1992).
- [2] M. Zhou, S. Sun, "Genetic algorithms: theory and applications", National Defence Industry Press, Beijing (1999).
- [3] G. Chen, X. Wang, Z. Zhuang, "Genetic algorithms and its applications", People's Posts & Telecommunications Publishing House, Beijing (1992).
- [4] J. Zhang, H. Chung, B. Hu, "Adaptive probabilities of crossover and mutation in genetic algorithms based on clustering technique", In: Proceedings of the 6th IEEE Congress on Evolutionary Computation (CEC'04), pp.2280-2287 (2004).
- [5] M. Srinivas, L. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms", IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, no. 4, (1994), pp. 656-66 .
- [6] H. Gao, B. Feng, L. Zhu, "Adaptive SAGA based on mutative scale chaos optimization strategy", In: Proceedings of the International Conference on Neural Networks and Brain (ICNN&B'05), pp. 517-520 (2005).
- [7] L. Chen, T. Zhang, M. Lv, "Chaos genetic algorithm for optimizing neural network-based state model for pipe network", Journal of Zhejiang University (Engineering edition), vol. 39, no. 6, (2005), pp. 874-877
- [8] B. Chen, X. Feng, "Self-adapting chaos-genetic hybrid algorithm with mixed congruential method", In: Proceedings of the 4th International Conference on Natural Computation (ICNC'08), (2008), pp.674-677.
- [9] M. Zhang, S. Zhao, X. Wang, "A hybrid self-adaptive genetic algorithm based on sexual reproduction and Baldwin effect for global optimization", In: Proceedings of the 11th IEEE Congress on Evolutionary Computation (CEC'09), (2009), pp.3087-3094.
- [10] Y. Zhuang, H. Yu, J. Xia, *et al*, "Optimization of linear induction machines based on a novel adaptive genetic algorithm", Journal of Southeast University (English Edition), vol. 25, no. 2, (2009), pp. 203-207.
- [11] J. Gao, M. Xiao, W. Zhang, "A rapid chaos genetic algorithm", In: Proceedings of the 1st International Conference on Swarm Intelligence (ICSI'10), Part I, LNCS 6145, (2010), pp.425-431 .
- [12] C. Dai, Y. Zhu, W. Chen, "Adaptive probabilities of crossover and mutation in genetic algorithms based on cloud model", In: Proceedings of IEEE Information Theory Workshop (ITW'06), (2006), pp. 710-713.

- [13] Y. Zhang, H. Zhang, Z. Liu, *et al*, "Crowding self-adaptive genetic algorithm and its application in facility layout problem", International Journal of Advancements in Computing Technology, vol. 4, no. 7, (2012), pp. 203-212
- [14] J. Huang, "Function optimization and image matching based on improved adaptive genetic algorithm", International Journal of Advancements in Computing Technology, vol. 4, no. 19, (2012), pp. 57-64.
- [15] Y. Duan, J. He, "Genetic algorithm and its improvement", In: Proceedings of the CSU-EPSA, (1998), pp. 39-52
- [16] M. Li, Z. Cai, G. Sun, "An adaptive genetic algorithm with diversity-guided mutation and its global convergence property", Journal of Central South University, vol. 11, no. 3, (2004), pp. 323-327.
- [17] L. Wang, T. Shen, Y. Zhao, "An improved adaptive genetic algorithm", Systems Engineering and Electronics, vol. 24, no. 5, (2002), pp. 75-78.
- [18] S. Shi, Q. Li, X. Wang, "Design optimization of brushless direct current motor based on adaptive genetic algorithm", Journal of Xi'an Jiaotong University, vol. 36, no. 12, (2002), pp. 1215-1218.
- [19] S. Ouyang, J. Wang, Z. Song, "A new improved genetic algorithm and its application", Journal of System Simulation, vol. 15, no. 8, (2003), pp. 1066-1073.
- [20] S. Ouyang, J. Wang, Y. Geng, "A new improved genetic algorithm", Computer Engineering and Applications, vol. 11, no. 4, (2003), pp. 13-15.
- [21] J. Jin, Y. Su, "An improved adaptive genetic algorithm", Computer Engineering and Applications, vol. 18, no. 6, (2005), pp. 64-69.
- [22] K. Mak, Y. Wong, X. Wang, "An adaptive genetic algorithm for manufacturing cell formation", International Journal of Manufacturing Technology, no. 16, (2000), pp. 491-497.
- [23] Q. Wu, Y. Cao, J. Wen, "Optimal reactive power dispatch using an adaptive genetic algorithm", Electrical Power & Energy Systems, vol. 20, no. 8, (1998), pp. 563-569.
- [24] Y. Yun, M. Gen, "Performance analysis of adaptive genetic algorithms with fuzzy logic and heuristics", Fuzzy Optimization and Decision Making, vol. 2, no. 2, (2003), pp. 161-175.
- [25] Y. Song, G. Wang, P. Wang, *et al*, "Environmental/economic dispatch using fuzzy logic controlled genetic algorithms", In: Proceedings of the International Conference on Generation, Transmission and Distribution, (1997), pp.377-382 .
- [26] C. Silva, "A survey of chaos and its applications", In: Proceedings of the IEEE MTT-S International Microwave Symposium Digest, (1996), pp.1871-1874.
- [27] Y. Ding, Y. Wang, "Application of genetic algorithm in image reconstruction", Communications Technology, vol. 131, no. 11, (2002), pp. 4-6.