A Restricted Genetic Algorithm Based on Ascending of Tangent Planes

Jinhua Zheng [1] **Yan Peng** [2]

[1] College of Information Engineering, Xiangtan University, Xiangtan, CN411105
jhzheng@xtu.edu.cn
[2] College of Information Engineering, Xiangtan University, Xiangtan, CN411105
pengyanpy@gmail.com

Abstract

Classical Genetic Algorithms（CGAs）accomplish the global search by selection, crossover and mutation. It has many shortages. Here we proposed a restricted genetic algorithm based on ascending of tangent plane (RGAATP). Our algorithm is simple and effective, which approaches the global solution step by step by ascending of the tangent planes. It is proved that the algorithm is global convergent, and it can solve the optimization of multimodal function [6] easily and effectively. The experimental results show that its performance is more stable and efficient than aSGA.

Keywords: restricted genetic algorithm, tangent planes, optimization of multimodal function.

1. Introduction

CGAs are based on the mechanics of artificial selection and genetic recombination operators. Most of the theories of genetic algorithms deal with the so-called schema theorem. But the crucial factors of the GA success are often not achieved [1]. And if the population size is small, probability selection can also generate some serious statistical errors. Furthermore, mutation process can also influence the stability of convergence process and its control parameters are very hard to choose. Most of the genetic algorithms used now are just the improved versions of the CGAs, which have many flaws, such as premature convergence, low convergence speed, bad stability in convergence and bad controllability in searching.

The previously proposed algorithms ([1] [2] [4] [5]) can not radically solve the drawbacks of CGAs, but only make improvements on some certain performances. But Restricted Genetic Algorithm ([6][7])（RGA）, which makes full use of the implicit parallelism of the CGAs and does the searching process with both lower selection probability and no mutation, has the advantages such as high efficiency of global search, good stability and controllability. But it is a local convergent method which can just solve the univariate problems with single-peak. The RGA combined with area searching [8] could solve the univariate problems with multi-peak and a part of multivariate problems with multimodal. Actually, the optimization problems in science and engineering are often the multivariate problems with multimodal. So the application of this algorithm is limited.

In this paper, we present the RGAATP which proposes a promising solution to solve the optimization of the multi-variant problems with multi-peak. Our algorithm is simple and effective. Due to the adoption of real-coding, the algorithm can decrease the length of the chromosome. It also solves the contradiction between the accuracy and the complexity of the computation. Generally the disadvantages of the CGA can be solved by RGAATP. Experiments show that our algorithm outperforms previous methods significantly.

The remainder of the paper is organized as follows. Section 2 describes the RGAATP algorithm, and Section 3 gives some theorems and proof. Then section 4 shows the experimental setup and results. Finally section 5 provides our conclusions.

2. RGAATP

Before discussing our algorithm, let us look at RGA first. RGA is a simple increment search procedure. The search procedure of RGA is similar to hill-climbing. It is proved that RGA is very effective for the problems with univariate and single-peak and it can converge to the local optima in a finite generation. Fig.1 gives pseudo-code for the RGA.

The domains in which new individuals generate in step 2 and step 4a are different. The domain in step 2 doesn't change in the whole process, but the domain in step 4a is changed continually. At the first time of running to the step 4a the domain is the neighborhood (Δ) of X . If X' is better than X , the domain is changed to be the neighborhood (Δ) of X' . Δ is a micro range which is used to control the extent of the search. So we can see that RGA is local convergent.

1. initialize
   $\mu = \theta$ /* control the the end of the algorithm*/
2. randomly generate initial population P
3. select the best solution X
4. while μ is not equal to zero do
   a. randomly generate population P'
   b. let X crossover with each solution in P'
   c. select the best solution X' from P'
   d. if(X' is better or equal to X) then
      i $X = X'$
      ii $\mu = \mu - 1$
      else $\mu = \theta$

Fig.1: The pseudo-code for RGA

In fig.1 parameter θ is the threshold value while $\mu = 0$ means the termination of the searching process in RGA. The crossover operation in RGA is shown below.

Definition1. Chromosome $X = (X_1, X_2, \cdots X_n)$, $X_i \in R$, $i = 1, 2, \cdots n$, where X_i denotes the *i*-th variable.

Crossover operation： Convex Combination is used in this paper, which is determined explicitly from the following formula:

$$C = A \otimes_{ram} B ,$$

$$C_i = \begin{cases} \alpha A_i + (1-\alpha) B_i & if \quad ram \le 0.5 \\ \alpha B_i + (1-\alpha) A_i & if \quad ram > 0.5 \end{cases}$$

where $\otimes_{ram}$ is the crossover operation, α and *ram* are randomly chosen from the interval [0,1] respectively.

Based on the this formula, different crossover operation is chosen according to the parameter *ram*. Moreover, if A and B are belong to domain D, C is also included in D. So this operation ensures that the process can search for the optima continuously in feasible region.

RGA is a important part of RGAATP. RGA in our algorithm is the same as the framework of Fig. 1. The pseudo-code of RGAATP is shown in Fig. 2.

1. initialize
2. randomly generate a initial solution X in domain D
3. while (X is smaller than local-elitist and ++count< *bignum*) do
   randomly generate a solution X
4. if (count!= *bignum*) then
   calling the RGA /*put the maximum in X 's peak into local-elitist */
   else end

Fig.2: The pseudo-code for RGAATP

An example of maximization problem is introduced to explain the algorithm. Let $X = (X_1, X_2)$ be bi-dimensional decision variable, F the fitness function which is also the objective function, and $D = [U_{11}, U_{12}] \times [U_{21}, U_{22}]$ the space of the decision variable.

bignum is a large positive value, which ensures the searching above the current tangent plane. Here Δ is equal to $[X_1 - \delta, X_1 + \delta] \times [X_2 - \delta, X_2 + \delta]$. δ is a positive real value which makes sure that $F(X_1, X_2)$ is single-peak in Δ. But U is a closed-range, it may result in $\Delta \not\subset U$ regardless of how small δ is. So if $\Delta \not\subset D$, $\Delta = \Delta \cap D$ according to the boundary of D.

At the beginning of the algorithm, $X \in D$ is randomly generated in step 2. The δ is made as small as possible so that $\Delta \subseteq D$ and the function has only a maximum in Δ. After calling the RGA, the maximum point $X_{\max 0} = (X_{1,\max 0}, X_{2,\max 0})$ and the maximum value $X_{3,\max 0} = F(X_{1,\max 0}, X_{2,\max 0})$ can get in Δ. The new coordinate plane $X_3 = X_{3,\max 0}$ replaces the original coordinate plane $X_3 = 0$. So the local maximum values which are less than $X_{3,\max 0}$ are turned to be negative values in the new coordinate space. Then they are no longer the maximum. Step 3 ensures that the following search is towards to the space above $X_{3,\max 0}$.The sketch map of the conversion is described in Fig.3.

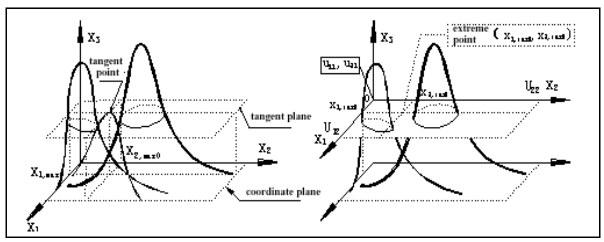

Fig.3: Left is the original coordinate plane where has three extreme points. Right is the new coordinate plane where has just two extreme points.

In this way, it would be much easier and more efficient to direct the search process to explore upper regions. The coordinate plane X_1X_2 moves toward above continuously and the extreme-value point which has smaller fitness value will be eliminated step by step. Finally the algorithm has the capability of global convergent.

3. Convergence in RGAATP

Here we will give the proof of global convergence. An algorithm is global convergent, if all of the following conditions are true.

- The algorithm is local convergent.
- The algorithm provides a method for shifting from current local optimum state to the unknown regions.
- The algorithm provides a strategy which can

control the search towards the whole space.

The proof of the convergence of RGAATP is given as follows.

THEOREM 1. *A RGAATP is global convergent.*

Proof.

(1) The algorithm satisfies the all conditions of global convergence.

① RGA is an algorithm which focuses on the single-peak problem. Obviously it is local convergent.

② Given domain $D=[U_{11},U_{12}]\times[U_{21},U_{22}]\times\cdots\times[U_{n1},U_{n2}]$, $F(X)=F(X_1,X_2,\cdots,X_n)$ is a multi-variant function which has m different extreme values assuming $extreme-v_1 < extreme-v_2 <\cdots< extreme-v_m$.

The RGA obtains the current local optimal solution $extreme-value_1 \geq extreme-v_1$ in the first generation. According to the current optimal solution which has just found we make a tangent plane. Then the $extreme-value$ point which is smaller than the obtained point is eliminated. In the following search the original value is set to be $(X_{e1},X_{e2},\cdots,X_{en})$ and $F(X_{e1},X_{e2},\cdots,X_{en}) > extreme-value_1$.

It is clearly that the searching process can transfer the current state space to another state space.

③ One situation: If there have some extreme values($extreme-v_{i1}$, $\cdots$, $extreme-v_{ik}$) which is equal to $extreme-v_i$, the algorithm is searching for the region corresponded to $extreme-v_i$ in fact.

The other situation: The search skips over the region corresponded to the middle extreme values. So the time of finding the local optimum is no more than m.

In the two situations the convergent speed is accelerated and the not-reached region is searched indirectly. So the RGAATP is capable of global searching.

(2) The algorithm terminates in finite time.

Assume that $F(X)=F(X_1,X_2,\cdots,X_n)$ has m different extreme values ($extreme-v_1$ < $extreme-v_2$ < $\cdots$ < $extreme-v_m$) whose corresponding domains are D_1, D_2, $\cdots$, D_m respectively. The extreme- $value_1$ will be obtained after first running of RGA. At the beginning of the RGA, the original solution $X_1 \in D_i$ ($i\in\{1,2,\cdots,m\}$) is randomly generated. So $extreme-value_1 \geq extreme-v_1$ (assuming that $extreme-value_1 = extreme-v_r$ for $r\in\{1,2,\cdots,m\}$). Then a new tangent plane at the point of $extreme-value_1$ is built in order to eliminate the extreme solutions which are less than $extreme-value_1$. Before calling the RGA for second times the solution $X_2 \in D_j$ ($j\in\{r+1,r+2,\cdots,m\}$) is generated at random and $extreme-value_2 \geq extreme-v_r \geq extreme-v_2$. So the RGAATP can find the globally maximal extreme point and the corresponding extreme value $extreme-v_\infty$ after calling the RGA for m times at the most.

4. Experiments

In this section, we empirically evaluate the RGAATP on multi-peak test problems to show its effectiveness in finding the optimum solution.

4.1. Experimental Setup

Test problems are chosen from a number of significant past studies in this area. We choose three multi-peak test problems, which are briefly explained below. And all of them are continuous.

Schaffer Function (SF):

$$f(x_1,x_2)=0.5-(\sin^2\sqrt{x_1^2+x_2^2}-0.5)/\left[1+0.001(x_1^2+x_2^2)\right]^2 \quad (1)$$

$$x_1,x_2\in(-4,4)$$

It has many local maxima but only one global maximum and the corresponding function value is 1.

Six-hump Camel Back Function (SCBF):

$$f(x_1,x_2)=(4-2.1x_1^2+x_1^4/3)x_1^2+x_1x_2+(-4+4x_2^2)x_2^2 \quad (2)$$

$$-3\leq x_1\leq 3,\quad -2\leq x_2\leq 2$$

It has six local minima and one global minimum whose corresponding function value is -1.031628.

Goldstein&Price's Function (GPF):

$$f(x_1,x_2)=(1+(x_1+x_2+1)^2\times(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)) \times(30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2))$$

$$-2\leq x_1,x_2\leq 2 \quad (3)$$

It has just one global minimum and the function value is 3.0000.

The change from minimized problem to maximized problem is very easy. So in this paper, all objective functions are to be maximized.

The advanced-SGA (aSGA) is slightly modified by simple genetic algorithm (SGA). The difference between aSGA and SGA is that aSGA has the strategy of preserving the current elitist individual. Obviously aSGA is global convergent. Here in all experiments, aSGA (binary coded) is executed in 200 generations per test and with a population size 20. The crossover rate is 0.75 and the mutation rate is 0.03.

Different parameter settings are used in different experiment of RGAATP (real coded). The crossover probability and the population size are the same as in aSGA. Here mutation is not utilized. In the first experiment, $bignum$ is 10000, $\mu=5$, δ is 0.0089. In the second experiment, $bignum$ is 500, $\mu=10$, δ is 0.1. And in the last experiment, $bignum$ is 40000, $\mu=10$, δ is 0.05.

Each algorithm was performed 100 times.

4.2. Performance Measure

For comparison, we use some metrics to evaluate the two algorithm's performance. They are convergent probability (CP), average running time (ART) and average value (AV) respectively. CP is $N/100$ where N is the times of finding the optimum. ART is the average CPU's running time. AV is S/100 where S is the sum of values got in each running of the algorithm.

4.3. Results

Table 2 and 3 show the experimental results for these three problems. In all problems, it is clear that the performance of RGAATP is better than that of aSGA. In SCBF, aSGA converges to the local optimum easily. The superiority of RGAATP for these three problems suggests that RGAATP is more powerful than other GAs in cases of multi-peak. Actually in RGAATP the GPF's result is not so good. In order to compare AV some parameters in RGAATP are changed to bigger. Obviously we can see that the result of RGAATP is better than aSGA. It is also important to stress that better results will be obtained by tuning of the parameters.

Table 2. The performance of RGAATP

	RGAATP		
	CP	ART(s)	AV
SF	0.32	0.0114846	0.9999979
SCBF	0.33	0.0019865	-1.03162
GPF	0.01	0.0311757	3.00017

Table 3. The performance of aSGA

	aSGA		
	CP	ART(s)	AV
SF	0	0.0550465	0.9927052
SCBF	0.15	0.0385949	-1.03066
GPF	0	0.0351047	3.00286

We do not make any serious attempt to find the best parameter setting for RGAATP. But now a second experiment is conducted in order to show the effect of different settings of P_{size}, $\mu(\theta)$, δ and $bignum$ on the RGAATP algorithm. Each different experimental condition involved selecting fixed values for P_{size}, $\mu(\theta)$, δ and $bignum$. GPF is adopted for the test problem. Here each set of parameters runs for 10 times. The results is described in Table 4.

We can see that P_{size}, $\mu(\theta)$ and δ exert greater effects on the speed and accuracy than $bignum$. Selecting each parameter is not so hard. Generally if the structure of the optimum problem is unkown, the setting of $bignum$, P_{size} and $\mu(\theta)$ should be larger but δ should be smaller in order to ensure the global convergence. However, in practice, depending upon the type of problem, different set of parameters may be required.

Table 4. The results of different parameter setting in RGAATP

	$bignum$	P_{size}	$\mu(\theta)$	δ	ART(S)	AV
1	100	20	5	0.0089	0.00332	3.00274
2	4000	20	5	0.0089	0.00451	3.00004
3	500	10	5	0.0089	0.00167	3.22984
4	500	30	5	0.0089	0.00209	3.00002
5	500	20	5	0.0089	0.00356	3.00004
6	500	20	10	0.0089	0.00241	3.00001
7	500	20	5	0.05	0.00294	3.00435
8	500	20	5	0.1	0.00116	3.00944

5. Conclusions

This paper has presented a novel RGAATP to solve optimization problems of multimodal functions. The experimental results have proved that it had many advantages such as fast convergent speed, good stability, good controllability and so on.

6. References

[1] Ling. Zhang, and Bo. Zhang, "The Statistical Genetic Algorithms," *Journal of Software*, Vol.8, No.5, 1997.

[2] Rudolph. G, "Convergence Analysis of Canonical Genetic Algorithms," *IEEE Transactions on Neural Networks*, 5(1), 1994.

[3] Chuanyu. Xu, "Hybrid Approach and its Generalization for Solving Premature Convergence of a Class of Genetic Algorithms," *Journal of Software*, Vol.9, No.3, 1998.

[4] Ming. Chen, "Optimization Computing Based on Evolution Genetic Algorithm," *Journal of Software*, Vol.9, No.11, 1998.

[5] Jinhua. Zheng, Zhenghua. Ye, Zuqiang. Meng, and Zixing. Cai, "The Genetic Algorithm based on Special Crossover," *Pattern Recognition and Artificial Intelligence*, 16(3), 2003.

[6] Jinhua. Zheng, and Zixing. Cai, "Restricted Genetic Algorithm Combined with Area Search," *Computer Engineering and Application*, Vol.36, No.1, 2000.

[7] Weihua. Zheng, and Jinhua. Zheng, "The Genetic Mechanism Analysis of Restricted Genetic Algorithm," *Natural Science Journal of Xiangtan University,* 25(2), 2003.

[8] Jinhua. Zheng, and Zixing. Cai, "Restricted Genetic Algorithm of Area Search based on Automatic Area Parting," *Computer Research and Development*, Vol.37, No.4, 2000.