

An Improved Evolutionary Algorithm with New Genetic Operation for Optimization Problem

Wang Jiekai, Hu Ruikai, Wang Chao*

College of Mathematics Science, Harbin Normal University,
Harbin, Heilongjiang Province, P.R. China

*Corresponding author, e-mail: hsdwangchao@126.com

Abstract

An improved evolutionary algorithm (SCAGA) is proposed in this paper for solving optimization problem. In order to control genetic operations in an effective range, the new algorithm regulate both of the crossover probability and mutation probability with the iteration process. In addition, SCAGA presents a new crossover strategy that restricts the cross of the chromosomes to some extent to protect good genes schema. We also perform the schema theorem on the algorithm process to analyze the working mechanism of SCAGA, and we conclude that the new algorithm is effective. According to experiment results for some test functions and TSP problems, SCAGA have a high performance in both constrained unconstrained optimization problems.

Keywords: evolutionary algorithm, crossover operator, mutation operator, crossover strategy, schema theorem

Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

Genetic algorithm (GA for short) is a kind of typical evolutionary algorithm [1]. It is a stochastic search algorithm for global optimization based on principle comes from evolution mechanism in nature. When GA is used to compare with other traditional algorithms, its unique coding pattern and searching method can often be more effective [2]. As a result, it is appropriate for optimization problem in complex system [3-5]. However, genetic algorithm also has some shortcomings such as local optimal solutions, lower convergence speed etc, which limit the use of genetic algorithm seriously. For such problem, many effective improvement for genetic algorithm have been proposed in recent years. And most research about GA focused on the genetic operator and the execution strategy [6-8].

The standard genetic operators actually provide a kind of stochastic search process with uncertain probability. While the operators can give each individual a chance for optimization, it also have a possibility lead into the recession of the group [9]. To reduce the blindness of genetic operations, an improved adaptive genetic algorithm is proposed in this paper. Meanwhile, the proposed algorithm use a new crossover strategy so as to make the algorithm more efficient.

The rest of this paper is organized as follows. In Section 2, we introduce the improved genetic algorithm SCAGA. In Section 3, we present a theoretical analysis use schema theorem for the algorithm. In Section 4, experimental results both on benchmark function and TSP problem are given and discussed. Finally, we conclude in Section 5.

2. Improved Adaptive Genetic Algorithm

2.1. Adaptive Genetic Algorithm

Due to the essence of evolution is a dynamic process, some researchers think related genetic parameters should be regulated adaptively rather than being invariable. The adaptive genetic algorithm (AGA) proposed by Srinvas could regulate crossover and mutation probability to get GA more efficient [10]. But it still has some disadvantages such as low convergence rate and high probability to local convergence. In recent years, many researchers in various fields try to find more efficient method to improve AGA. Some comprehensive methods could be found in

[11-12]. And there are also some works based on heuristic mechanism turn out be efficient [13-14].

In this paper, we present a new improved adaptive genetic algorithm SCAGA. The proposed new algorithm is based on adjusting mutation and crossover probability to regulate the evolution process. Before iteration, SCAGA use a new initialization method to make sure that the initialized population can distribute in the solution space evenly. And the algorithm could optimize crossover and mutation probability dynamically by improved crossover and mutation operators. A series of new methods to regulate the genetic operators dynamically were applied so as to improve the performance of the proposed algorithm.

2.2. Population Initialization

In general, way of genetic algorithm for population initialization is completely random. Randomly initial population tends to make individuals unevenly distributed in the solution space. It is easy to cause the imbalanced evolution of the algorithm in the iteration process and result in local optimal solutions. In order to overcome this problem, we use a new method with uniform solution space to make the initial individuals can be evenly distributed.

Step 1: Divide the solution space averagely into N subspaces according to the population size.

Step 2: Each subspace generate sub-individuals with completely random way.

Step 3: Compose all the sub-individuals and get the initial population.

The individuals of initial population consist of $A(0)=\{A_1(0)^1, A_2(0)^1, \dots, A_n(0)^1\} \cup \{A_1(0)^2, A_2(0)^2, \dots, A_n(0)^2\} \cup \dots \cup \{A_1(0)^N, A_2(0)^N, \dots, A_n(0)^N\} \in \mathbb{R}^N$. This approach can improve the diversity of population on the premise of initializing population randomly. And it can improves the convergence performance of proposed algorithm.

2.3. The Improvement of Crossover and Mutation Operator

According to the schema theorem [1], the nature of the genetic algorithm should be the replacement of the original schema and the formation of excellent schema. In the process of genetic operation, it should keep a new schema as much as possible in the early time of population evolution, in order to maintain population diversity. In the later period of population evolution, it should try to maintain an appropriate mode and prevent the destruction, to prevent the algorithm from prematurity. Individuals have a greater probability of high adaptability to contain the fine schema, so they are more suitable for the relatively small crossover and mutation probability during the evolution. On the contrary, low fitness individuals not only is unlikely to contain fine schema, but they may have a possibility to damage the fine schema by hybridizing with individuals contained fine schema. Therefore, the low fitness are more suitable for smaller crossover probability. For the almost average fitness, we can increase their crossover and mutation probability properly to explore the potential fine schema in these individuals.

In this paper we uses formula (1) as follow that decreases progressively while the algorithm iterating:

$$x = \sin\left(\left(1 - \frac{gen}{gen_{max}}\right) \cdot \frac{\pi}{2}\right) \quad (1)$$

Where gen represent the current evolution generation, gen_{max} represent the preset total evolution generation.

More formally, let N be the population scale, a be the percentage that a particular individual fitness accounts for the sum of all individual's, F and F_{avg} represent the individual fitness and average fitness of population.

Apparently, when a is closer to $1/N$, we can see the particular individual fitness F is closer to F_{avg} , thus $(1+Na)$ is closer to 2. According to the mentioned above, at this time the crossover probability should be higher. Conversely, when the F is more far from F_{avg} , namely $(1+Na)$ is closer to 1, the crossover probability should be smaller. The following formula are used in this paper:

$$y = \ln(1 + N \cdot a) \quad (2)$$

In order to regulate the crossover probability and mutation probability in an effective range, we set up following updating formula for cross probability according to the *sigmoid* function:

$$z = x \cdot y = \sin\left(\left(1 - \frac{gen}{gen_{max}}\right) \cdot \frac{\pi}{2}\right) \cdot \ln(1 + N \cdot a) \quad (3)$$

$$p_c' = \frac{1}{1 + e^{-z}} \quad (4)$$

Therefore, the crossover probability of two individuals is:

$$p_c = \frac{p_{c_1}' + p_{c_2}'}{2} \quad (5)$$

Because mutation operator can disturb the performance of the algorithm to a certain extent, the determination of mutation probability is very important. In general, the selection of mutation probability is mainly got according to experience, so its reliability is often suspectable. According to the following formula, the mutation probability could be adjusted dynamically in a better way.

$$p_m = \left| \sqrt{\sin\left(\left(1 - \frac{gen}{gen_{max}}\right) \cdot \frac{\pi}{2}\right) - 0.5} \right| \quad (6)$$

Where p_m represent the current mutation probability.

The value of p_m will decrease gradually with the iteration process. In this way, it can avoid getting into the local optimum in the early period, and it can also improve the convergence performance of SCAGA in the later period.

2.4. The Improvement of Crossover Strategy

Crossover operator also has a great influence on the convergence rate of GA. However, traditional crossover operation often has a certain blindness. The offspring individuals which are formed by crossover of parental chromosomes may discard or destruct the fine genes schema existed in parental individuals. Therefore, in addition to the adjustable crossover and mutation probability, this paper also presents a new crossover strategy based on our related works [15-16]. So that the fit schema existed in parental generation is protected to transfer to offspring as far as possible.

Definition 1 (Coding Distance): Suppose that two individuals are encoded in binary respectively, and the coding length is L , we say the coding distance of X and Y is:

$$\varepsilon_{X,Y} = \sum_{m=1}^L (k_m \times c_m) \quad (7)$$

Where k is the given weight, $c_m = \begin{cases} 1, & a_{X,m} = a_{Y,m} \\ 0, & a_{X,m} \neq a_{Y,m} \end{cases}$

Definition 2 (Coding similarity): The coding similarity between two individuals X and Y can be computed with formula (8).

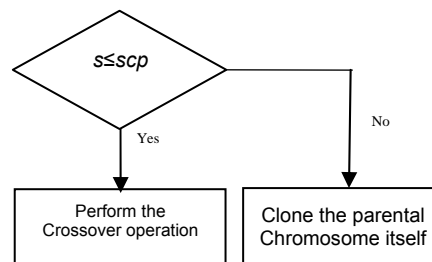
$$s = S(X, Y) = \varepsilon_{X,Y} / \sum_{m=1}^L k_m \quad (8)$$

Obviously, the value of coding similarity of any two individuals is between $[0, 1]$. In order to control the crossover operation, here we use a critical value as reference.

Definition 3 (Standard Crossover Point): The standard crossover point (scp) is a critical value to decide whether the crossover operation should be implemented, and it can be computed as follows:

$$scp = \frac{gen_{max} + 2\sqrt{gen}}{3gen_{max}} \quad (9)$$

From the formula (9), scp will continuously increase with the iteration of genetic algorithm. Based on above definition, the new crossover strategy can be described as:



The value of coding similarity between individuals is relatively lower in the prior stage of SCAGA. In order to ensure that the outstanding genetic schema will not be broken for the full evolution of chromosome group, scp value should be relatively low controlled crossover operation. On the contrary, in the late stages of SCAGA, the difference between individuals can be very small, so the scp shall also increase. According to formula (9), the intersection of the dynamic control standards can help improve the efficiency and convergence performance of the algorithm.

2.5. Improved Adaptive Genetic Algorithm

The main operations of the proposed algorithm SCAGA can be summarized as follows:

Step 1: Initialize population, initial population consist of $A(0) = \{A_1(0)^1, A_2(0)^1, \dots, A_n(0)^1\} \cup \{A_1(0)^2, A_2(0)^2, \dots, A_n(0)^2\} \cup \dots \cup \{A_1(0)^N, A_2(0)^N, \dots, A_n(0)^N\} \in R^N$;

Step 2: Evaluation of the fitness value of each individual;

Step 3: Selection operator: Perform selection operator and get $A(t) = \{A_1(t), A_2(t), \dots, A_{n \times N}(t)\}$;

Step 4: Crossover operator: Perform the crossover operation and get $A'(t) = \{A'_1(t), A'_2(t), \dots, A'_{n \times N}(t)\}$, if the crossover operation condition is satisfied, where the crossover probability can be calculated according to formula (3)(4) and (5);

Step 5: Mutation operator: Perform the mutation operation according to formula (6) and turn individuals into $A''(t) = \{A''_1(t), A''_2(t), \dots, A''_{n \times N}(t)\}$;

Step 6: Repeat Step3~ Step5 until a stopping criterion is satisfied;

Step 7: Output the best solution of all individuals.

3. The Schema Theorem Analysis for SCAGA

The schema theorem has been a efficient method for analyzing genetic algorithm. The schema theorem predicts that growth of high fitness has a good effect on algorithm process, and indicates show crossover and mutation effect on the propagation of genetic schema. In this section, we analyze proposed algorithm with schema theorem. The symbols used is as follows.

| | |
|-------------|-------------------------------------------|
| H | : a present schema |
| $f(H)$ | : the fitness value of schema |
| \bar{f} | : the average fitness value of population |
| \bar{f}_H | : the average fitness value of schema H |

$m(H, gen+1)$: the number of expected offspring created in generation $gen+1$ of schema H

$m(H, gen)$: the number of schema H in generation $gen+1$

$\delta(H)$: the defining length of schema H

$O(H)$: the order of schema H

l : the length of the encoded solution

$M(H, gen)$: the number of solution of generation gen of schema H

Suppose that the selection operator have implemented, we here to consider the effect of crossover operator and mutation operator.

According to the schema theorem, the number of expected offspring created in generation $gen+1$ of schema H is given by:

$$m(H, gen+1) \geq m(H, gen) \frac{f(H)}{\bar{f}} \left[1 - p_c \frac{\delta(H)}{l-1} \right] \quad (10)$$

The parameter P_c is given by (5), (6) and (7).

To simplify analysis procedure, we assume that P'_{c1} is equals to P'_{c2} , therefore, after substituting for P_c in formula (10), we could get:

$$m(H, gen+1) \geq m(H, gen) \frac{f(H)}{\bar{f}} \left[1 - \frac{1}{1 + e^{-\sin\left(\left(1 - \frac{gen}{gen_{max}}\right) \cdot \frac{\pi}{2}\right) \cdot \ln(1+N \cdot a)}} \frac{\delta(H)}{l-1} \right] \quad (11)$$

Now we could get the $M(H, gen)$, and to estimate its value, we consider to acquire the summation of $m(H, gen)$ over all solutions.

$$M(H, gen+1) = \sum_{i=1}^{M(H, gen)} m(H, gen+1) \quad (12)$$

From formula (12), we could get:

$$M(H, gen+1) \geq \sum_{i=1}^{M(H, gen)} m(H, gen) \frac{f(H)}{\bar{f}} \left[1 - \frac{1}{1 + e^{-\sin\left(\left(1 - \frac{gen}{gen_{max}}\right) \cdot \frac{\pi}{2}\right) \cdot \ln(1+N \cdot a)}} \frac{\delta(H)}{l-1} \right] \quad (13)$$

From the inequality (13), we get:

$$M(H, gen+1) \geq \sum_{i=1}^{M(H, gen)} \frac{f(H)}{\bar{f}} \left[1 - \frac{1}{1 + e^{-\sin\left(\left(1 - \frac{gen}{gen_{max}}\right) \cdot \frac{\pi}{2}\right) \cdot \ln(1+N \cdot a)}} \frac{\delta(H)}{l-1} \right] \quad (14)$$

Consider that $\sum_{i=1}^{M(H, gen)} f(H) = M(H, gen) \times \bar{f}_H$, we can modify (14) as:

$$M(H, gen+1) \geq M(H, gen) \frac{\bar{f}(H)}{\bar{f}} \left[1 - \frac{1}{1 + e^{-\sin\left(\left(1 - \frac{gen}{gen_{max}}\right) \cdot \frac{\pi}{2}\right) \cdot \ln(1+N \cdot a)}} \frac{\delta(H)}{l-1} \right] \quad (15)$$

Now consider the crossover strategy proposed in (9), (10) in section 2.4,

We could rearrange terms of formula (15) as follows:

$$\begin{aligned}
M(H, gen+1) &\geq M(H, gen) \frac{\bar{f}(H) (gen_{\max} + 2\sqrt{gen})}{af} \\
&\quad - M(H, gen) \frac{\bar{f}(H)}{af} \frac{gen_{\max} + 2\sqrt{gen}}{1 + e^{-\sin((1-\frac{gen}{gen_{\max}})\frac{\pi}{2})\ln(1+N\cdot a)}} \frac{\delta(H)}{3gen_{\max}(l-1)}
\end{aligned} \tag{16}$$

Formula (16) represents the schema theorem influenced by proposed crossover operator and crossover strategy in SCAGA.

After that we could simply get form of the schema theorem of SCAGA effected by mutation operation as follows due to mutation probability used in (8).

$$\begin{aligned}
M(H, gen+1) &\geq M(H, gen) \frac{\bar{f}(H) (gen_{\max} + 2\sqrt{gen})[1 - O(H)p_m]}{af} \\
&\quad - M(H, gen) \frac{\bar{f}(H)}{af} \frac{gen_{\max} + 2\sqrt{gen}}{1 + e^{-\sin((1-\frac{gen}{gen_{\max}})\frac{\pi}{2})\ln(1+N\cdot a)}} \frac{\delta(H)[1 - O(H)p_m]}{3gen_{\max}(l-1)}
\end{aligned} \tag{17}$$

Since $p_m = \left| \sqrt{\sin\left(\left(1 - \frac{gen}{gen_{\max}}\right) \cdot \frac{\pi}{2}\right)} - 0.5 \right|$, the schema theorem for SCAGA could be generalized to:

$$\begin{aligned}
M(H, gen+1) &\geq M(H, gen) \frac{\bar{f}(H) (gen_{\max} + 2\sqrt{gen})[1 - O(H)] \left| \sqrt{\sin\left(\left(1 - \frac{gen}{gen_{\max}}\right) \cdot \frac{\pi}{2}\right)} - 0.5 \right|}{af} \\
&\quad - M(H, gen) \frac{\bar{f}(H)}{af} \frac{gen_{\max} + 2\sqrt{gen}}{1 + e^{-\sin((1-\frac{gen}{gen_{\max}})\frac{\pi}{2})\ln(1+N\cdot a)}} \frac{\delta(H)[1 - O(H)] \left| \sqrt{\sin\left(\left(1 - \frac{gen}{gen_{\max}}\right) \cdot \frac{\pi}{2}\right)} - 0.5 \right|}{3gen_{\max}(l-1)}
\end{aligned} \tag{18}$$

Since the schema theorem for SGA can be described as follows:

$$m(H, gen+1) \geq m(H, gen) \frac{f(H)}{f} \left[1 - p_c \frac{\delta(H)}{l-1} - O(H)p_m\right] \tag{19}$$

Consider (13), we get:

$$M(H, gen+1)_{SGA} \geq \sum_{i=1}^{M(H, gen)} m(H, gen) \frac{f(H)}{f} \left[1 - p_c \frac{\delta(H)}{l-1} - O(H)p_m\right] \tag{20}$$

To compare $M(H, gen+1)_{SCAGA}$ and $M(H, gen+1)_{SGA}$, we could divide (18) by (20), or we could simplify the computation as follows:

$$\begin{aligned}
\frac{M(H, gen+1)_{SCAGA}}{M(H, gen+1)_{SGA}} &\geq \frac{(1 - p_{c_{IAGA}} \frac{\delta(H)}{l-1})(1 - O(H)p_{m_{SCAGA}})}{(1 - p_{c_{SGA}} \frac{\delta(H)}{l-1})(1 - O(H)p_{m_{SGA}})} \\
&\geq \frac{1 + p_{c_{SCAGA}} p_{m_{SCAGA}} \frac{\delta(H)}{l-1} O(H) - p_{c_{SCAGA}} O(H) - p_{m_{SCAGA}} \frac{\delta(H)}{l-1}}{1 + p_{c_{SGA}} p_{m_{SGA}} \frac{\delta(H)}{l-1} O(H) - p_{c_{SGA}} O(H) - p_{m_{SGA}} \frac{\delta(H)}{l-1}} \\
&\approx \frac{p_{c_{IAGA}} p_{m_{IAGA}} - p_{c_{IAGA}} - p_{m_{IAGA}}}{p_{c_{SGA}} p_{m_{SGA}} - p_{c_{SGA}} - p_{m_{SGA}}}
\end{aligned} \tag{21}$$

After substituting for P_m and P_c of SCAGA in formula (21), we could get:

$$\frac{M(H, gen+1)_{SCAGA}}{M(H, gen+1)_{SGA}} \geq 1 \quad (22)$$

Equivalently, from (22) we can immediately get:

$$M(H, gen+1)_{IAGA} \geq M(H, gen+1)_{SGA} \quad (23)$$

Because the probability of solution disruption with high fitness can be smaller than solution with lower fitness, we could observe that proposed SCAGA use high fitness value to promote schema. Meanwhile, SCAGA could also get schema increase steadily. From (23), we can see SCAGA is more efficient than SGA according to the schema theorem.

4 Simulations

4.1. Function Test

In order to test the performance of SCAGA, three commonly used multimodal test function are performed in this section. Both functions and their variable range are summarized as follows:

$$\min f_1(x, y) = (4 - 2.1x^2 + x^4/3)x^2 + xy + (-4 + 4y^2)y^2$$

When the range of f_1 is $x \in (-100, 100)$, $y \in (-100, 100)$, the optimum is -1.031628.

$$\min f_2(x, y) = \left\{ \sum_{i=1}^5 i \cos[(i+1)x + i] \right\} \cdot \left\{ \sum_{i=1}^5 i \cos[(i+1)y + i] \right\}$$

When the range of f_2 is $x \in (-10, 10)$, $y \in (-10, 10)$, the optimum is -186.7309.

$$\max f_3(x, y) = 0.5 - (\sin^2 \sqrt{x^2 + y^2} - 0.5) / (1 + 0.001(x^2 + y^2))^2$$

When the range of f_3 is $x \in (-100, 100)$, $y \in (-100, 100)$, the optimum is 1.

$$\min f_4\left(\frac{-\sin^3(2\pi x_1) \sin(2\pi x_2)}{x_1^3(x_1 + x_2)}\right)$$

When the range of f_4 is $x_1 \in (0, 10)$, $x_2 \in (0, 10)$, and the constrained condition is:

$$g_1(x) = x_1^2 - x_2 + 1 \leq 0, \quad g_2(x) = 1 - x_1 + (x_2 - 4)^2 \leq 0,$$

The optimum is $x^* = (1.2279713, 4.2453733)$, $f(x^*) = -0.095825$.

$$\min f_5(-(\sqrt{n})^n \prod_{i=1}^n x_i)$$

When the range of f_5 is $n=10$, $x_i \in (0, 1)$, $i=1, \dots, n$, and the constrained condition is:

$$h_1(x) = \sum_{i=1}^n x_i^2 - 1 = 0, \quad \text{the optimum is } x^* = 1/\sqrt{n} (i=1, \dots, n).$$

$$\min f_6(-(100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2) / 100))$$

When the range of f_6 is $x_i \in (0, 10)$, $i=1,2,3$, and the constrained condition is:

$$g(x) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leq 0, \quad p, q, r = 1, 2, \dots, 9$$

The optimum is $x^* = (5, 5, 5)$, $f(x^*) = -1$.

To evaluate the performance of the proposed algorithm, SGA, AGA and PSO is used for comparisons. We tested above functions for 50 times to give a comparison of average solution and optimal solution for these four algorithms,

The optimization results are listed in Table 1.

Table 1. The Tested Comparison Table of SGA, AGA, PSO and SCAGA for Test Functions

| Function | SGA | | AGA | | PSO | | SCAGA | |
|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | optimal | mean | optimal | mean | optimal | mean | optimal | mean |
| f_1 | -0.9622 | -0.9453 | -0.9934 | -0.9681 | -1.0314 | -0.1012 | -1.0312 | -1.0229 |
| f_2 | -186.7250 | -186.7010 | -186.7300 | -186.7180 | -186.7280 | -186.7100 | -186.7300 | -186.7240 |
| f_3 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 |
| f_4 | -0.0958 | -0.0966 | -0.0958 | -0.0967 | -0.0958 | -0.0947 | -0.0958 | -0.0958 |
| f_5 | -1.0000 | -0.9866 | -1.0000 | -0.9972 | -1.0000 | -1.0000 | -1.0000 | -1.0000 |
| f_6 | -1.0000 | 0.9466 | -1.0000 | 0.9878 | -1.0000 | -0.9997 | -1.0000 | -1.0000 |

From Table 1, the optimization results of SCAGA for f_{1-6} , including both of constrained and unconstrained optimization problem, are better than those in SGA and AGA and PSO algorithm [10]. For f_1 (Camel function), compared the mean and optimal results with SGA and AGA and PSO algorithm, except the results with the mean solution of PSO is slightly better than those of SCAGA, other related simulation results of the proposed algorithm are better than other algorithm for comparison.

We use f_2 (Shubert function) and f_3 (Schaffer function) to test convergence performance of SCAGA. The comparison of average results of SGA, AGA, and SCAGA are listed in Table 2.

Table 2. Comparing of Convergence Performance of SGA, AGA, and SCAGA

| Algorithm | Function | Population | Time | Generation |
|-----------|----------|------------|------|------------|
| SGA | f_2 | 100 | 48 | 58 |
| | f_2 | 200 | 50 | 55 |
| | f_3 | 300 | 42 | 96 |
| | f_3 | 500 | 45 | 93 |
| AGA | f_2 | 100 | 50 | 54 |
| | f_2 | 200 | 50 | 52 |
| | f_3 | 300 | 43 | 92 |
| | f_3 | 500 | 44 | 88 |
| SCAGA | f_2 | 100 | 50 | 50 |
| | f_2 | 200 | 50 | 49 |
| | f_3 | 300 | 42 | 89 |
| | f_3 | 500 | 47 | 87 |

We use SGA, AGA, and SCAGA to test Schaffer function and Shubert function independently for 50 times. Form the comparison result from Table 2, the convergence times of SCAGA are more than SGA and SCAGA while the convergence generation in average of SCAGA is less than SGA and AGA.

Form the experiment above, we can see SCAGA has a better convergence performance than SGA and SCAGA, and can obtain better optimal solution than other compared algorithm.

4.2. TSP Problem Test

The traveling salesman problem (*TSP*) has attracted attention from many mathematicians and computer scientists as a *NP*-complete problem. It involves finding the shortest Hamiltonian cycle in a complete directed graph, and it is a good ground to test performance of optimization algorithm.

Firstly we choose to test 30-city and 105-city *TSP* problem to compare preference of SCAGA with both SGA and AGA.

Table 3. Performance of SGA, AGA, and SCAGA for TSP

| Cities | Average | | | Optimal | | | Genes | Pop. Size |
|------------|---------|---------|---------|---------|-----|-------|-------|-----------|
| | SGA | AGA | SCAGA | SGA | AGA | SCAGA | | |
| 30(424.0) | 442.1 | 430.2 | 428.1 | 0 | 7 | 9 | 100 | 1000 |
| 105(14383) | 16344.3 | 14801.4 | 14689.3 | 0 | 4 | 4 | 500 | 2000 |

We use population size as 1000 and 2000 for 30-city and 105-city *TSP* respectively. For both problems, SCAGA represent a better performance than AGA and SGA on the average of tour length. And SCAGA located the optimal solution for 9 times in 30-city problem and 4 times in 105-city problem, while AGA located the optimal solution for 7 times in 30-city problem and 4 times in 105-city problem.

Secondly, we select 8 *TSP* instances to evaluate the effectiveness of SCAGA. Figure 1 is a comparison with different algorithm for the percentage deviations of the average solution to the best known solution.

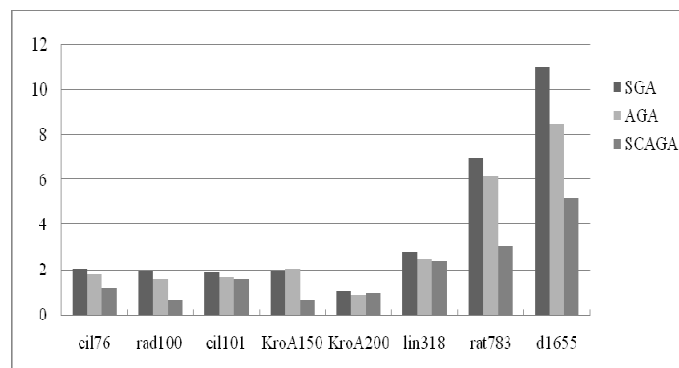


Figure 1. Percentage Deviations of the Average Solution in each *TSP* Dataset of Three Algorithms

As we can see in Figure 1, the proposed SCAGA gets relatively smaller percentage deviations than both SGA and AGA. That indicate SCAGA is a efficient algorithm for optimization problem.

5. Conclusion

In this paper, we present a new evolutionary algorithm SCAGA based on the improved genetic operation, which adopts a dynamic method to regulate crossover and mutation probability. Solution of GA can often be infected by its randomly initializing population and genetic parameter. The improvement in this paper could make the SCAGA obtain greater optimization performance and convergence performance. Meanwhile, we propose a new

crossover strategy to determine whether to take crossover operation. The dynamic control for the crossover operation could help improve the convergence rate of the algorithm. According to the schema theorem analysis for SCAGA, we could see that SCAGA is more efficient than SGA. In Section 4, we also compare the optimization performance of SCAGA with several typical algorithm in 6 different test functions and *TSP* problems. Therefore, it can be concluded that SCAGA is highly improved and is efficient to solve both constrained and unconstrained optimization problem.

References

- [1] JH Holland. *Adaptation in Natural Artificial Systems*. MIT Press. 1975: 1-17.
- [2] Goldberg DE. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison Wesley. 1989.
- [3] Chung-Cheng Lu, Vincent F Yu. Data envelopment analysis for evaluating the efficiency of genetic algorithms on solving the vehicle routing problem with soft time windows. *Computers & Industrial Engineering*. 2012; 2(63): 520-529.
- [4] A Serdar Tasan, Mitsuo Gen. A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Computers & Industrial Engineering*. 2012; 3(62): 755-761.
- [5] José A Castellanos-Garzón, Fernando Díaz. An evolutionary computational model applied to cluster analysis of DNA microarray data. *Expert Systems with Applications*. 2013; 7(40): 2575-2591.
- [6] D Thierens. *Adaptive mutation rate control strategies in genetic algorithms*. Proceedings of the 2002 Congress on Evolutionary Computation CEC. 2002
- [7] MM Lotfi, R Tavakkoli-Moghaddam. A genetic algorithm using priority-based encoding with new operators for fixed charge transportation problems. *Applied Soft Computing*. 2013; 5(13): 2711-2726.
- [8] L Cordella, C De Stefano, F Fontanella, A Marcelli EvoGeneSys. A new evolutionary approach to graph generation. *Applied Soft Computing*. 2013; 4(13): 1922-1938.
- [9] David H Wolpert, William G Macready. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*. 1997; 4(1): 67-82.
- [10] M Srinivas, L Patnaik. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithm. *IEEE Trans. On Systems, Man, and Cybernetics*. 1994; 24(4): 656-666.
- [11] Licheng Jiao, Lei Wang. A Novel Genetic Algorithm Based on Immunity. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*. 30; 5.
- [12] Claudio Comis, Da Ronco, Ernesto Benini. A Simplex Crossover based evolutionary algorithm including the genetic diversity as objective. *Applied Soft Computing*. 2013; 4(13): 2104-2123.
- [13] Siva Venkadesh, Gerrit Hoogenboom, Walter Potter and Ronald McClendon. A genetic algorithm to refine input data selection for air temperature prediction using artificial neural networks. *Applied Soft Computing*. 2013; 5(13): 2253-2260.
- [14] Adam L Chen, Daniel H Martinez. A heuristic method based on genetic algorithm for the baseline-product design. *Expert Systems with Applications*. 2012; 5(39): 5829-5837.
- [15] Jiekai Wang. *Improved Genetic Algorithm Based on the Small Group Parallel*. International Workshop on Future Communication and Networking. IWFCN. 2011.
- [16] CAI Mingdi, YAO Huanmin, WANG Jiekai, etc. *A Novel Evolutionary Algorithm with Improved Genetic Operator and Crossover Strategy*. The 2013 2nd International Conference on Information Technology and Management Innovation. ICITMI. 2013.