ISSN:0975-9646

Suvarna Patil et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 4669-4673

Comparison and Analysis of Different Mutation Strategies to improve the Performance of Genetic Algorithm

Suvarna Patil[#1], Manisha Bhende[*2]

[#1]*PG Student, Computer Department, Pune University*

[*2]*Research Scholar, Computer Department, Pune University*

DYPIET, Pune, Maharashtra, INDIA

***Abstract*— Genetic algorithm (GA), as an important intelligence computing tool, is a wide research content in the application domain and the academic circle now. This paper elaborates the improvement of premature convergence in GA used for optimizing multimodal numerical problems. Mutation is the principle operation in Genetic Algorithm (GA) for enhancing the degree of population diversity, but it is proved that it is not efficient often, mostly in traditional GA. The mutation rate is a tradeoff between computing time and accuracy. This paper presents a comparative analysis of different mutation approaches, based on the distributions for the purpose to examine their performance, evaluate the average improvement of chromosomes and investigate their ability to find solutions with the high precision. The proposed approach consists of mainly three components. The first component describes simple genetic algorithm with the problem of genetic algorithm. The second component elaborates on different mutation strategies of genetic algorithm. These strategies improves the performance of genetic algorithm which promotes and enriches the existing intelligent optimization theory and methods, and have a wide application prospect in optimization of complex systems, production management and other fields. In this paper, we have compared Dynamic Mutation Genetic Algorithm (DMGA), Schema Mutation Genetic Algorithm(SM-GA) algorithm, Compound Mutation (BCM-GA) algorithm, Clustered based adaptive mutation (CBAM) algorithm, Hyper Mutation Based Dynamic Algorithm (HMDA).**

***Index Terms*— Genetic algorithm, Mutation**

I. INTRODUCTION

Genetic Algorithm (GA), proposed by Holland in 1975, is a kind of search optimization algorithm based on the theory of evolution and the genetic mutation theory of Mendel. Recently, it has been a hot research in many fields such as data mining, optimization control, artificial intelligence etc., and the wide applications have been achieved in many corresponding fields. However, the theory of genetic algorithm is still imperfect, and its main performance in the premature convergence of genetic algorithm. Leading to premature convergence of genetic algorithm is mainly due to lack of effective gene[1-3]. Genetic algorithms comprise the basic operation of encoding, selection, crossover and mutation, the core theory is the schema theorem. Respectively, a variety of genetic manipulation simulates the basic characteristics of biological evolution from different angles in Genetic algorithms, including the mutation which is an important measure in the prevention of local convergence. Many researchers have improved the genetic algorithm from different perspectives for the premature convergence problem. For example, in the literature [4] the author put forward the basic strategy that using different crossover and mutation on different species group that are classified by the size of their fitness. In the literature [5] the author resolved the premature convergence problem by adjusting the probability of the crossover and mutation. In the literature [6] the author improved the algorithm convergence speed by adjusting the probability of the various individual's crossover and mutation. In the literature [7] the author put forward the concept of degrees of different groups, and automatically adjusts the size of the mutation rate according to the degrees of different groups in order to ensure the diversity of groups and avoid the premature phenomenon. One common feature of these methods is that they use the larger mutation rate to prevent the premature convergence of the algorithm, but with increasing of the mutation rate, the solving process of the genetic algorithm would gradually turn into random search, then make the algorithm lose the real terms of bionics, if the mutation rate increases when the species appear to be immature, the algorithm would destroy the obtained better information. From the intuitive sense, in the solving process of genetic algorithm, if the relative good model can be determined by the timely adoption of the current information of the species group, and protect these models in the following genetic manipulation , then such operation strategy must be able to enhance the performance of algorithm[8]. A genetic algorithm consists of three principal operations:

i. The crossover operation generates the offspring from ***two*** chosen individuals in the population, by exchanging some bits of the **two** individuals. The offspring will then inherit some characteristics of their parents.

ii. The mutation operation generates the offspring by randomly changing one or several bits of individuals the offspring may then possesses different characteristics from their ascendants. Mutation can then avoid local search in the searching space and increase the probability of finding the global optimum.

iii. The selection operation chooses some of offspring to survive according ***to*** some predefined rules. The number of the population is then under good control.

II. MUTATION STRATEGIES

In this section different mutation strategies are studied with their features and drawbacks.

A. Dynamic Mutation Genetic algorithm (DMGA):

Dynamic Mutation Genetic Algorithm (DMGA) [9] automatically chooses an appropriate mutation operator, or to handle situations in which different operators are suitable for different genetic stages. The Dynamic Mutation Genetic algorithm uses several mutation operators simultaneously to generate next generation. Initially, the mutation ratios of all the available mutation operators are equal and are set to l/N of the total mutation ratio for the problem. Each mutation operator is then applied according to its assigned probability, and the fitness of offspring is evaluated. The mutation operators which result in higher average fitness values then have their control rates increased. The mutation operators that result in lower average fitness values then have their control rates decreased. Finally, the most suitable mutation operator stands out and controls almost all the mutation action in the population. The detailed algorithm is described as follows:

The Dynamic Mutation Genetic Algorithm:

Step 1: It defines a suitable representation of the problem to be solved.

Step 2: It creates an initial population of N individuals for evolution.

Step 3: It defines a suitable fitness function f with which to evaluate individuals.

Step 4: Apply crossover operator to generate the possible offspring.

Step 5: And then assign several appropriate or common candidate mutation operators, each with an initial mutation ratio.

Step 6: It performs genetic operations to generate possible offspring (each mutation operator uses its mutation ratio).

Step 7: Then evaluate the fitness value of each individual.

Step 8: After that calculate the average growth value Progress generated by each mutation operator. Assume parent p is chosen by a mutation operator to produce a child a.

Step 9: It selects the superior N individuals according to their fitness values.

Step 10: It adjust the mutation ratios of the candidate mutation operators according to the average growth values.

Step 11: Then, if the termination criterion is not satisfied, go to Step 4, otherwise, stop the algorithm.

When the given termination criterion is satisfied, the individual with the highest fitness value is the output as the best solution.

B. Schema Mutation Genetic Algorithm (SMGA):

In order to avoid falling into local optimum, mutation operation of the genetic algorithm is taken as an operation strategy, its aim is to constantly produce greater individual that fitness is much better. The mutation operator in the following is commonly used at present [10]:

i. **Basic mutation operator:** Change each individual gene value according to a certain probability. The method is easy for operating, but it cannot effective control to mutation result.

ii. **Non-uniform mutation:** Mutation probability is greater early in the evolution, with the evolution advancing mutation probability appropriate to reduce.

iii. **Adaptive Mutation:** An individual with high fitness correspond to a smaller mutation probability, and individuals with low fitness corresponds to a high mutation probability. This method can effectively protect the excellent individuals, but it is easy to fall into local convergence.

Based on the above analysis, it is proposed [10] a complex mutation strategy based on schema mutation from the point of protecting the existing excellent individuals: the method can promote a better individual producing.

Step 1 According to a certain proportion, count the common feature of the individuals of higher fitness in current population, that is, count the same gene loci, and then get the relative quasi-optimality schema *H*.

Step 2 In the process of generating the next generation, some individuals belong to schema *H*, advance mutation operation in accordance with the smaller probability $P_c^{(1)}$, but some individuals do not belong to schema *H*, it will advance mutation operation in accordance with the larger probability $P_c^{(2)}$.

The genetic algorithm with the above-mentioned compound mutation strategy as the genetic algorithm based on schema mutation, for short SM-GA is given. It is clear that SM-GA is usually GA when $Pc^{(1)} = P_c^{(2)}$. Therefore SM-GA is the promotion of the existing GA, for specific purposes, in the following discussion, other genetic operation of SM-GA is as follows:

1) Selection operator: Adopting proportional selection operator in contemporary population, the probability that the i^{th} individual is selected.

2) Crossover operator: Adopting single-point crossover operator. That is to exchange the two individual genes after the cross-bit on a cross-bit of randomly selecting basis.

C. Compound Mutation Genetic Algorithm (BCM-GA):

The mutation operation is the important strategy used to avoid trapping into the local optimum, it generally aims to gradually generate individuals with even bigger fitness, so that it leads to evolution of the whole population, the role is shown as follows[11]:

1) Seek for better solution according to contemporary population, 2) Keep the diversities of given population, and also assure the further evolution of population and avoid local convergence. In the specified mutation process, various mutation operators have different searching inclination, the smaller mutation intensity are fit for local search, the bigger mutation intensity is fit for global search. For real coding the most commonly used mutation operation is as follows:

(1) Uniform mutation: In this for the variable x, the mutated result is $x = a + r(b - a)$, and [a, b] denotes the

range of x, r is random number in [0, 1]. These are a completely random variation, and mostly it lack of targeted. (2) Gauss mutation: It is a commonly used operator, widely used with the genetic algorithm. This type of mutation is a mutation operation to focus near a mutation individual from the characteristics of normal distribution, is known from the process of Gauss mutation. The mutation may lead to infeasible solutions.

A complex mutation strategy is proposed to protect the existing excellent individuals. The main idea of this mutation is: the individuals of larger fitness have a smaller mutation and coincidentally the individuals of smaller fitness have a larger mutation. Its illogical meaning is that the individuals of optimal are protected through a small mutation probability and individuals of sub-optimal has a larger mutation probability. The method will expand users search space and increase the diversity of the individuals. The algorithm is given as,

Step 1: Mutation criteria function:

Mutation criteria function is a method reflecting bio-mutation process. Let m(x) be mutation criteria function and X= (x1, x2, xn) be the individual for mutation. pm be given mutation probability. The execution rule is that processing the following operation bit by bit from the first bit,

1) Randomly generate number r ϵ [0, 1]
2) If r ≥ pm, then xi doesn't mutate.

Step 2: Compound mutation strategy:

In this section, starting from the structural angle, establish compound mutation strategy based on mutation criteria function. Using sorting the fitness of individuals in each generation, the individuals of the higher value of the objective function compose optimal Filial-population of the generation. The individuals of optimal Filial-population properly reduce mutation intensity by changing mutation function. Mutation intensity increases to individuals which do not belong to optimal Filial-population. This mutation strategy will retain excellent individual of each generation, and also promote the generation of excellent individual.

D. Clustered Based Adaptive Mutation (CBAM) Algorithm:

The performance of GA is always sensitive to the definition of parameters, such as rates of crossover and mutation, population size, and so on. Hence, many researchers concentrate to the studies of robust GA, for instance, for the Binary-coded GA (BGA), the adaptive GA (AGA) whose probability of mutation rate is varied depending on the fitness values of the solution, the solutions with high fitness are protected while solutions with under average fitness are all disrupted. In [14], authors suggested the large-size mutation that decrease the mutation range and descend the mutation rate exponentially could obtain better performance for GA optimization. Besides, for the real-coded GA (RGA), Haupt suggested a small population size with relatively large mutation rate is far superior to the large population size with low mutation rate [15]. Unfortunately, these adaptive mutation schemes could not support a suitable and reliable rate instantaneously. Moreover, the problem of the survival rate of mutated offspring was ignored often, so the redundant computation would come from great mutation number [12].

For many applications, however BGA is most natural to use an alphabet of many characters or real numbers to form chromosomes. RGA have also been developed that use integer or real-valued representations and order-based representations where the order of variables in a chromosome plays an important role. From with several empirical comparisons, the real-valued encodings presents better performance than binary encodings. BGA consumes more computation time to encode and decode chromosomes unceasingly in every generation while it transforms between genotype and phenotype to evaluate the fitness. From view of chromosome motion of GA, it is known that the chromosomes converged by crossover and spread by mutation, respectively. Because of survival of fittest in selection operation, the convergent force is always more powerful than spreading, a concept of degree of population diversity was proposed to quantitatively characterize and theoretically analyze the problem of premature convergence in GA within the framework of Markov chain. It had proved that the degree of population diversity converges to zero with probability one should decrease the search ability of a TGA and lead the occurrence of premature convergence.

Now, a concept to observe convergent situation by clustering and offspring survival rate is considered to understand the cause of premature convergence on TGA. Here, the survival offspring means an offspring at last generation and is selected to be current parent. In many numerical optimization simulations by TGA, discovered some main shortcomings encountered in evolutionary process.

According to analysis mentioned above, in given work the improvement of survival rate of mutated offspring is the main goal. The density of population is regards as a principle feature for this work, therefore, introduced a cluster method, nearest neighborhood which has low computation cost and without predefining cluster number, that is suitable to handle the problem. After clustering, the redundant and repeat parent will be discarded for enhancing offspring competitiveness in selection operation. Moreover, mutated offspring fill back to parents that are thrown out, so the mutation number is tuned adaptively and automatically by degree of population diversity.

The algorithm is given below,

Step 1: Predefine the parameters to generate initial population and evaluate their fitness values. *For i=1, 2,*

Step 2: Generate crossover offspring by (1) or (2)

Step 3: Clustering parent chromosomes and remove redundancies.

Step 4: Renew the parents by mutated offspring.

Step 5: Select new population from crossover offspring and renewed parent chromosomes which include mutated offspring.

Step 6: **end** until the terminated condition is satisfied.

E. *Hyper Mutation Based Genetic Algorithm (HMGA):*

In stationary environments, convergence at a proper pace is really what is expected for GAs to locate the optimum solutions for many optimization problems. However, for Dynamic Optimization Problem (DOP) s, convergence usually becomes a big problem for GAs because changing environments usually require GAs to keep a certain population diversity level to maintain their adaptability. Hyper-mutation [13] is also a basic technique that enables genetic algorithms to cope with the dynamic environments. In this[13], it uses well-designed hyper-mutation GAs to solve the Dynamic Multicast Routing Problem (DMRP) in MANETs. In traditional genetic algorithm, the mutation rate is fixed during the entire evolutionary process. However, the core idea in hyper-mutation is to adjust the mutation rate adaptively. Since mutation can help to generate new solutions, intuitively, when environmental changes occur, the mutation rate should be increased to a high level to maintain the population diversity. When the environment is in a steady state, the mutation rate should be decreased to a low level to guarantee that the population can exploit the good search space.

Following the principle, it is proposed two different types of hyper-mutation schemes,

1. high low hyper-mutation
2. gradual hyper-mutation

In the high low hyper-mutation, a new concept of change interval is defined. A change interval refers to the number of generations between two consecutive changes. For the first half of the change interval, the mutation rate is set at the predefined high level and for the second half, the mutation rate is set at the pre-defined low level. In the gradual hyper-mutation, when there is a change to the environment, the mutation rate is increased to the pre-defined high level. Then at each generation the mutation rate is gradually decreased till it reaches the predefined low level or another change arrives. The idea is clear. When the environmental change occurs, the current population faces the biggest challenge and a high mutation rate can help it to jump out of the local optimum. After that, the capability of the population in coping with the new environment becomes stronger and stronger, the mutation rate should be set lower and lower.

III. Analysis Of Different Mutation Strategies

In this section, we have analyzed five mutation strategies in accordance with their features, advantages and disadvantages.
It is given as follows,

Sr. No	Name	Features	Advantages	Disadvantages
1.	Dynamic Mutation Genetic Algorithm (DMGA)	Uses Several Mutation Operators	Automatically choose appropriate mutation operator	Premature convergence problem.
2.	Schema Mutation Genetic Algorithm (SM-GA)	Deals with premature convergence problem for genetic algorithm.	It have a good wide range of constitutive property and maneuverability, also are superior to ordinary genetic algorithm on computational Efficiency and convergence stability.	Problem of local optimum
3	Compound Mutation Genetic Algorithm (BCM-GA)	It is the specific strategy used to avoid trapping into the local optimum.	1) Seek for better solution on the basis of contemporary population, 2) Keep the diversities of population, assure the further evolution of population and avoid local convergence.	It is not possible to produce suitable mutation number.
4	Clustered Based Adaptive Mutation Algorithm (CBAM)	It produces suitable mutation number adaptively and it guaranteed the higher survival rate of offspring in selection operation.	Cluster-based GA choose a suitable mutation number for reducing computing time and maintain the population variety for preventing premature convergence.	Problem of random search
5	Hyper Mutation Based Dynamic Algorithm (HMDA)	The hyper-mutation scheme adaptively adjusts the mutation rate during the run of GAs. The population diversity of GA can be maintained to the changing environment.	Used two different types of hyper-mutation schemes to enhance GA's performance in dealing with the dynamic environment.	The hyper-mutation GA where the mutation rate is gradually increased is not suitable for the problem.

IV. CONCLUSION

In this paper we have presented a comparative analysis of different mutation-based algorithms. The Dynamic Mutation algorithm automatically applies several mutation operators in solving problem. It uses more one mutation operators to generate next generation. The SM-GA have wide range of constitutive property and maneuverability, also are superior to ordinary genetic algorithm on computational efficiency and convergence stability. These discussions promote and enrich the existing intelligent optimization theory and methods, and have a wide application prospect in optimization of complex systems, production management and other fields. The Compound Mutation algorithm is based on analyzing the shortcomings of the existing genetic algorithm, we implement different mutation to individuals of every generation from structural angle, and set up a class of broad operational genetic algorithm. Clustered Based Adaptive Mutation Algorithm (CBAM) produces suitable mutation number adaptively and it guaranteed the higher survival rate of offspring in selection operation.

REFERENCES

[1] J. H. Holland, “Genetic algorithms and the optimal allocations of trials”, SIAM Journal of Computing, Vol. 2, pp. 88–105, 1973.

[2] W. Atmar, “Noteson the Simulation of Evolution”, IEEE Trans. On SMC, Vol. 24, No. 1, pp. 130-147, 1994.

[3] D. B.Foge, “An Introduction to Simulated Evolutionary Optimization”, IEEE Trans.on SMC, Vol. 24, No. 1, pp. 3-14, 1999.

[4] A. S. Jiao, “Improvement and Convergence Analysis of Genetic Algorithm”, Machinery Research and Application, Vol. 21, No. 4, pp. 90-92, 2008.

[5] S. X. Tang, and Z.Chen, “Research on Improved Genetic Algorithm”, Computer and Modernization, Vol. 2008, No. 12, pp. 12-14, 2008.

[6] C. M. Huang, and X. X. Chen, “Improvement of Genetic Algorithm to Xiao Sheng Mirror”,Journal of Beijing Institute of Technology, Vol. 24, No. 8, pp. 675-678,2004.

[7] J. Zhuang, and S. A.Wang, “Self- regulate genetic algorithm research”,Journal of System Simulation, Vol. 36, No. 11, pp. 281-282,2003.

[8] W. M. Hui, and Y. G.Xi, “Genetic algorithm global convergence and analysis to computational efficiency”, Control Theory and Applications, Vol. 13, No. 4, pp. 455-460, 1996.

[9] Hong, Tzung-Pei, and Hong-Shung Wang. "A dynamic mutation genetic algorithm." *Systems, Man, and Cybernetics, 1996., IEEE International Conference on*. Vol. 3. IEEE, 1996.

[10] Li, Fachao, and Tingyu Zhang. "Study on Genetic Algorithm Based on Schema Mutation and Its Performance Analysis." *Electronic Commerce and Security, 2009. ISECS'09. Second International Symposium on*. Vol. 1. IEEE, 2009.

[11] Li, Fachao, Tingyu Zhang, and Chenxia Jin. "A kind of genetic algorithm based on compound mutation strategy and performance study." *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on*. IEEE, 2009.

[12] Sun, Tsung-Ying, et al. "Cluster-based adaptive mutation mechanism to improve the performance of genetic algorithm." *Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on*. Vol. 1. IEEE, 2006.

[13] Cheng, Hui, and Shengxiang Yang. "Hyper-mutation Based Genetic Algorithms for Dynamic Multicast Routing Problem in Mobile Ad Hoc Networks." *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*. IEEE, 2012.

[14]S. Deutsch, “The case for large-size mutations,” IEEE Transactions on Biomedical Engineering, vol. 48, Issue: 1, pp. 124-127, 2001 [13] W.-N. Chen and J.