An Improved Adaptive Genetic Algorithm and its application to image segmentation

Wang Lei, Shen Tingzhi

(Dept. of Electronic Engineering, Beijing Institute of Technology, Beijing, 100081, China)

Abstract—Genetic Algorithm (GA) is derived from the mechanics of genetic adaptation in biological systems, which can search the global space of certain application effectively. The proposed algorithm introduces three parameters, fit_{max} , fit_{min} , and fit_{ave} to measure how close the individuals are, so as to improve the Adaptive Genetic Algorithm (AGA) proposed by M. Srinivas. At the same time, the elitist strategy is employed to protect the best individual of each generation, and Remainder Stochastic Sampling with Replacement (RSSR) is employed in the proposed IAGA to improve the basic reproduction operator.

The proposed IAGA is applied to image segmentation. The experimental results exhibit satisfactory segmentation and demonstrate the learning capabilities of it. By determining p_c and p_m of the whole generation adaptively, it strikes a balance between the two incompatible goals: “sustain the global convergence capacity” and “converge rapidly to global optimum”.

Key words—Genetic Algorithm; Adaptive Genetic Algorithm; crossover probability; mutation probability; image segmentation

I. INTRODUCTION

Genetic Algorithm has been established as a heuristic method to optimization problems especially in a complex and big search space. The work presented here introduced an Improved Adaptive Genetic Algorithm (IAGA) based on Simple Genetic Algorithm (SGA) and Adaptive Genetic Algorithm (AGA) for optimization problems.

The remainder of this paper is organized as follows: Section II contains a brief overview of GA and discusses its problems. Following this review, Section III describes an AGA proposed by M. Srinivas including analysis to it. In Section IV, we proposed our improvement to AGA called IAGA. Then the proposed IAGA is applied to image segmentation problem in Section V. Section VI presents the experimental results and compares the results of IAGA with that of SGA. Finally, Section VII provides the conclusions of this paper.

II. OVERVIEW OF GENETIC ALGORITHMS

Genetic Algorithms were pioneered at the University of Michigan by John Holland and his associates^[1]. The term “Genetic Algorithm” is derived from the fact that its operations are loosely based on the mechanics of genetic adaptation in biological systems. Genetic Algorithms can be briefly characterized by two main concepts: a Darwinian notion of fitness or strength which determines an individual’s likelihood of affecting future generations through reproduction; three genetic operators known as “reproduction”, “crossover”, and “mutation” in which “reproduction” produces new individuals by combining selected members of the existing population while “crossover” and “mutation” create new offspring based on the structure of their parents.

In SGA the gene is two-valued, represented by a bit. The following genetic operators are used in a search procedure: proportional reproduction, one-point crossover, and simple mutation.

The reproduction operator in SGA known as “proportional reproduction” decides the expected number of times an individual is selected for recombination, which is proportional to its fitness relative to the rest of the population.

Let N be the population size, fit_i be the i^{th} individual’s fitness, the possibility for this i^{th} individual is

defined as

$$p_i = \frac{fit_i}{\sum_{i=1}^N fit_i} \quad (i = 1, 2, \dots, N)$$

An obvious drawback of “proportional reproduction” is that individuals with high fitness may not be selected and reproduced especially when the generation turns to be relatively small. The selection error is caused by the random selection techniques.

One-point crossover employed in SGA chooses a point at random, called the crossover point, and exchanges the segments to the right of this point.

Simple mutation is related to a bit (gene) that could be changed according to the mutation probability.

III. ADAPTIVE GENETIC ALGORITHM (AGA) PROPOSED BY M. SRINIVAS^[2]

p_c and p_m (probabilities of crossover and mutation) are increased when the population tends to get stuck at a local optimum, and decreased when the population is scattered in the solution space. p_c and p_m are defined as followed:

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

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

Where fit_{ave} is average fitness value of the population; $fit_{\max}$ is maximum fitness value of the population; fit' is the larger of the fitness values of the solutions to be crossed; fit is the fitness values of an instance (solution) of schema. $0 < k_1, k_2, k_3, k_4 \leq 1$, k_3 is the default crossover probability, and k_4 is the default mutation probability.

In M. Srinivas' algorithm, p_c and p_m are varied depending on the fitness value of each solution. High-fitness solutions are ‘protected’, while solutions with subaverage fitnesses are totally disrupted. However, M. Srinivas' algorithm adjusts p_c and p_m according to individual's behavior, which is liable to lead to the lack of collaboration. Thus this algorithm is liable to get stuck at a local optimum in some particular cases (e.g., stagnant period of the GA). Another drawback of M. Srinivas' algorithm lies in the fact that it requires computational time for each individual which has negative effects on program's efficiency, especially in hardware implementation.

IV IMPROVED ADAPTIVE GENETIC ALGORITHM (IAGA)

A. Adaptive p_c and p_m (crossover probability and mutation probability)

The proposed algorithm is not concerned with the evolution of a single individual, but instead is concerned with the evolution of the whole group. Three parameters, $fit_{\max}$, $fit_{\min}$, and fit_{ave} , are introduced to measure how close the solutions are.

The closeness of $fit_{\min}$ and $fit_{\max}$ indicates how close the entire generation is. The closer they are, the more possible the population gets stuck at a local optimum. We regard the entire generation “locally convergent” when $\frac{fit_{\min}}{fit_{\max}} > b$ where $0 < b < 1$. The equation is easy to be satisfied when b is close to 0. Thus the entire generation is easy to be “locally convergent”.

The closeness of fit_{ave} and $fit_{\max}$ indicates how disperse the individuals are. We regard each individual “concentrated with each other” when $\frac{fit_{ave}}{fit_{\max}} > a$ [4] where $0.5 < a < 1$. The equation is easy to be satisfied when a is close to 0.5. Thus each individual is easy to be thought as “concentrated with each other”.

When both $\frac{fit_{ave}}{fit_{\max}} > a$ and $\frac{fit_{\min}}{fit_{\max}} > b$ are fulfilled, the generation is considered convergent, and

then p_c and p_m are adapted by its convergent degree. Or else, p_c and p_m keep as same as default setting.

$$p_c = \begin{cases} p_c \frac{1}{1 - \frac{fit_{\min}}{fit_{\max}}} & \frac{fit_{ave}}{fit_{\max}} > a, \frac{fit_{\min}}{fit_{\max}} > b \\ p_c & \text{Otherwise} \end{cases}$$

$$p_m = \begin{cases} p_m \frac{1}{1 - \frac{fit_{\min}}{fit_{\max}}} & \frac{fit_{ave}}{fit_{\max}} > a, \frac{fit_{\min}}{fit_{\max}} > b \\ p_m & \text{Otherwise} \end{cases}$$

B. The Elitist Strategy

In the IAGA, the elitist strategy^[5] is employed by which the best individual is copied into the succeeding generation to protect the best solutions from the high level of disruption.

C. Improvement to Basic Reproduction Operator

Remainder Stochastic Sampling with Replacement (RSSR)^[5] is employed in the proposed IAGA to improve the basic reproduction operator. RSSR reproduction operates as follow steps:

1) Let N_i be the expected number of times an individual is selected for recombination, which may possibly not be an integer.

$$N_i = \frac{N \cdot fit_i}{\sum_{i=1}^N fit_i} \quad (i = 1, 2, \dots, N)$$

2) Let $\lfloor N_i \rfloor$ be the integer part of N_i so as to construct $\sum_{i=1}^N \lfloor N_i \rfloor$ individuals of the next offspring.

3) Replace each individual's fitness by its new fitness which is calculated as $fit_i - \frac{\lfloor N_i \rfloor \cdot \sum_{i=1}^N fit_i}{N}$. Then,

get the left $N - \sum_{i=1}^N \lfloor N_i \rfloor$ individuals for the next offspring by applying the simple proportional reproduction.

By combining random selection techniques and certain selection techniques, RSSR reproduction assures individuals with higher fitness than average fitness being selected for recombination, thus turns to have less selection error.

V IAGA'S APPLICATION TO IMAGE SEGMENTATION

Image segmentation is typically the first, and most difficult, task (also known as an ill-defined problem) of any automated image understanding process. It refers to the grouping of parts of an image that have "similar" image characteristics. Thresholding of the gray-level values is a popular tool used in image segmentation. In this paper, the IAGA is used to locate the optimal threshold in a gray-level image, where the pixels with lower gray level than threshold are regarded as object, and pixels with higher gray level than threshold are regarded as background.

There are two major considerations in practical application of GA: one is how to map the optimization problem into GA's search space, i.e., how to encode the individuals so as to solve the optimization problem. The other one is how to choose proper fitness function.

The image segmentation deals with pixels of different gray-level. Since the practical image used in this paper is 256-level, each individual in the population consists of 8 bits.

The fitness function is derived from Otsu method, which is defined as

$$fit(t) = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 = \omega_0\omega_1(\mu_0 - \mu_1)^2$$

Where t is the threshold, ω_0 is the probability of object which is assumed to be dark pixels, μ_0 is the mean of object's gray level, ω_1 is the probability of background which is assumed to be bright pixels, μ_1 is the mean of background's gray level.

Optimal threshold t^* is

$$t^* = \underset{t \in \{0,1,2,\dots,255\}}{\text{Arg Max}} fit(t)$$

There are still a lot of debates as to the best combination of GA parameter settings. Schaffer et al.^[6] suggest the parameter settings to be:

Population size 20-30

Crossover probability 0.75-0.95

Mutation probability 0.005-0.1

In the IAGA, the population size is 30, k_3 (default crossover probability) is 0.8, k_4 (default mutation probability) is 0.01, predetermined number of generations is 40. Natural evolution of the population continues until the predetermined number of generations is reached or the optimal threshold of each generation remains same for 10 generations.

An abstract procedure of the Improved Adaptive Genetic Algorithms is given below, where $N(k)$ is a population of candidate solutions to a given problem at generation k :

- 1) $k = 0$, generate an initial population $N(k)$.
- 2) Evaluate population $N(k)$ in terms of the IAGA.
- 3) Perform the IAGA's reproduction, crossover, and mutation.
- 4) Finish the IAGA and get the final optimal threshold if termination condition is satisfied, else return to 2).

VI EXPERIMENTS AND RESULTS

A. Performance Measure

It is possible for GA to converge to a threshold close to real optimal threshold due to its inherent character. We introduce a parameter, err , to measure the closeness between the GA's threshold and real optimal threshold. When

$$\frac{|t - t^*|}{t^*} \leq err$$

is fulfilled, we think GA has converged to a satisfactory threshold. In our case, err is 0.06.

Since experimental result is possibly different each time due to GA's convergence ability, we employ the average value to measure the SGA and IAGA's performance by performing the experiment repeatedly (in our case, 100 times).

We introduce 4 measures: average computational time (Time), average optimal threshold (Threshold), average number of generations to obtain global optimal threshold (Generations), and the number of runs for which the GA gets stuck at a local optimum (Stuck), i.e., fail to locate the global optimal threshold. The images in the experiments are 256 gray-level with the size of 256×256 .

B. Experimental Results

FIG. 1. gray-level image of "PORTRAIT": (a) original image; (b) Otsu method (global optimal threshold=111); (c) SGA (average threshold=110.48); (d) IAGA (average threshold=111).

FIG. 2. gray-level image of “CHINESE PAINTING”: (a) original image; (b) Otsu method (global optimal threshold=124); (c) SGA (average threshold=122.06); (d) IAGA (average threshold=124.2).

TABLE 1
COMPARISON OF PERFORMANCE OF SGA AND IAGA FOR “PORTRAIT” IMAGE

	Time	Threshold	Generations	Stuck
SGA	0.06049 s	110.48	10	11
IAGA	0.0669	111	11.68	0

TABLE 2
COMPARISON OF PERFORMANCE OF SGA AND IAGA FOR
“CHINESE PAINTING” IMAGE

	Time	Threshold	Generations	Stuck
SGA	0.06549	122.06	10.15	15
IAGA	0.0655	124.2	10.42	1

We measured the performance of the IAGA segmentation relative to that of SGA. SGA produced the optimal threshold in 10~16% of the experimental runs while IAGA determined the optimal threshold in only 0~2% of the experimental runs using the same data.

C. Analyses to experimental results

Once an image segmentation criterion is chosen, the optimal threshold can be determined accurately by it. The task for any optimization algorithm is to converge to this threshold as rapidly as possible.

To “converge rapidly to global optimum” and “sustain the global convergence capacity” are two twin goals in any optimization algorithm, but unfortunately they are considered incompatible. When it comes to GA, The diversity of individuals in the population may be disrupted when GA tries to “converge rapidly to global optimum”, hence cause “premature convergence”. On the opposite, convergence speed may not be assured if GA tries to “sustain the global convergence capacity”. A good algorithm is supposed to strike a balance between the two incompatible goals.

SGA employs static p_c and p_m , which is one of the reasons for that SGA fail to promise convergence speed or global convergence in some cases. The IAGA takes both “converge rapidly to global optimum” and “sustain the global convergence capacity” into account. p_c and p_m are increased when the IAGA tends to converge to a local optimum and decreased when a population scatters in the solution space. On the other hand, the adaptation to p_c and p_m is imposed on the entire population so as to enable the population to head for the global optimum in harmony.

VII. CONCLUSION

We have developed an Improved Adaptive Genetic Algorithm (IAGA) based on Simple Genetic Algorithm (SGA) and Adaptive Genetic Algorithm (AGA). Though based on AGA, IAGA is not concerned with the evolution of a single individual, but instead is concerned with the evolution of the whole group. In IAGA, the settings of p_c and p_m are adjusted automatically depending on the evaluation. The elitist strategy and RSSR reproduction are also employed to improve its capability.

The IAGA is applied to segment a gray-level image where Otsu method’s criterion is adapted to determine the optimal threshold. Experiments are conducted to evaluate the performance of IAGA and their results show that the IAGA yield better segmentation than the SGA.

Through the IAGA is applied to specific problem domain of image segmentation, it is hoped that the IAGA architecture will provide a basis for the development of both high performance and robust Genetic Algorithm.

REFERENCES

- [1] Holland, J. H. Adaptation in Natural and Artificial System. Ann Arbor: The University of Michigan Press, 1975
- [2] M. Srinivas, L. M. Patnaik. Adaptive probabilities of crossover and mutation in Genetic Algorithm. IEEE Transactions On Systems, Man and Cybernetics, 1994, 24(4): 656~667
- [3] De Jong K A. An analysis of the behavior of a class of genetic adaptive systems:[Ph.D Dissertation]. University of Michigan, No.76-9381, 1975
- [4] Ma Junshui, Liu Guizhong, Jia Yulan. The Big Mutation: An Operation to Improve the Performance of Genetic Algorithms. Control Theory and Applications. 1998, 15(3): 404~408
- [5] Brindle A. Genetic algorithms for function optimization:[Ph.D Dissertation]. University of Alberta, 1981.
- [6] J. D. Schaffer, R. A. Caruana, L. J. Eshelman, and R. Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. Proc. 3rd Int. Conf. Genetic Algorithms, 1989, 51~60