Journal of University of Science and Technology Beijing
Vol.7(2000), No.1, p.72

Generalized Self-Adaptive Genetic Algorithms

Bin Wu[1,2], *Xuyan Tu*[1], *Jian Wu*[2]

1) Information Engineering School, University of Science and Technology Beijing, Beijing 100083, China
2) Department of Information and Control Engineering, Southwest Institute of Technology, Mianyang 621002, China

(Received 1998-11-27)

Abstract: In order to solve the problem between searching performance and convergence of genetic algorithms, a fast genetic algorithm-generalized self-adaptive genetic algorithm (GSAGA) is presented. (1) Evenly distributed initial population is generated. (2) Superior individuals are not broken because of crossover and mutation operation for they are sent to subgeneration directly. (3) High quality immigrants are introduced according to the condition of the population schema. (4) Crossover and mutation are operated on self-adaptation. Therefore, GSAGA solves the coordination problem between convergence and searching performance. In GSAGA, the searching performance and global convergence are greatly improved compared with many existing genetic algorithms. Through simulation, the validity of this modified genetic algorithm is proved.

Key words: generalized self-adaptive; genetic algorithm; initial population; immigration; fitness function

In genetic algorithms, convergence is realized through reproduction and searching performance is achieved through crossover and mutation. Some individuals with high average fitness survive and reproduce through evolution, and finally develop into super-individuals, which cause closed-competition and close breeding, slow down or even cease the process of evolution. This may result in the convergence of the local optimal solution too soon. Mutation operation widens the searching area in order to find the global optimum, but the increase of mutation effects the speed of convergence. The existing modified genetic algorithms help solve the problems between searching performance and convergence, but the approximate global optimum can only be reached on the condition of many iterative steps, which causes difficulties in practice. So the Generalized Self-Adaptive Genetic Algorithm (GSAGA) is raised for the problem in this paper.

1 Principles of GSAGA

1.1 Generate initial population

In many modified genetic algorithms, initial population is randomly generated, which neglects how the initial population is distributed in the solution space. Some individuals may integrate in certain area. It is disadvantageous to widen the searching space and converge on the global optimum. In GASGA, certain distance or equal distance between individuals is required in generating initial population, so that the initial individuals are evenly distributed in the solution space.

Definition The different corresponding bits of two numeric string of the same length using a as its radix is defined as general hamming distance, signaled as GH.

Assume the individual is based on a as its radix, k as the length, N as the size of the population. Then the GH of the individuals in the initial population should be required as

$$\mathrm{GH}_{ij} \geq (k-b), (i \neq j) \tag{1}$$

where i, j stand for two individuals ($i,j = 1,2,\cdots,N$); b is a constant, decided by the codes. When the code is a binary string, $b = \mathrm{int}(k/2)$. If the code is a decimal string, $b \geq 2$.

The population size N influences the efficiency of the algorithms. When N is too small, the algorithms will be inefficient or can't reach the solution. While N is too big, then convergence takes too much time. So, assume

$N = 3k, (k > 4)$.

An a-based character string of length k has a^k code string. There are $a^{k-(k-b)+1}$ code string whose GH is greater than or equal to $(k-b)$ in the a^k code string. So in GSAGA, the size of population $N = 3k$ is far less than $a^{k-(k-b)+1}$. This means GH $\geq (k-b)$ can be fully achieved in the initial population.

GH is the general hamming distance of the two character strings of length L. $2^{L-\mathrm{GH}}$ is the amount of the same schema contained in the two character strings and 2^L is the schema of the character string of length L. So the different schema of the two character string is $(2^L - 2^{L-\mathrm{GH}})$. The greater the GH is, the more will be the types of schema between character strings and the schema of the population will be increased as well. If the initial population is generated in this way, individuals can be evenly distributed in the solution space. The distinct individual difference and abundant schema of the initial population can be guaranteed. So this algorithm has better chance to locate the local optimal solution.

1.2 Fitness function

In genetic algorithms, statuses of individuals are evaluated by fitness function. The calculation of fitness could be either simple or complicated. It depends on optimization itself. For some problems, only a mathematical formula is needed; for some problems without such formula, they can be resolved through a series of inference procedures based on rules; for some other ones, the combination of the mentioned methods is used.

A mathematical formula is adopted to calculate the fitness in GSAGA. The object function of optimization is assumed as $J(x)$, and fitness function as $f(x)$, then

$$f(x_i)=J(x_i)-J_{\min}+\frac{J_{\max}-J_{\min}}{N} \tag{2}$$

where,

$$J_{\min}=\min_{i=1}^{N}J(x_i),\ J_{\max}=\max_{i=1}^{N}J(x_i).$$

1.3 Reproduction operator

To make sure the best individual obtained will not be broken because of the selection, crossover and mutation operators, $0.1N$ (10%) superior individuals with higher fitness in father generation population are directly delivered to the subgeneration and turned into individuals of it. The left $0.9N$ (90%) individuals are sorted in ascending order according to their fitness. Assume the corresponding individual sorting number is r_i ($i=1,2,\cdots,0.9N$), then the individuals are reproduced as

$$\operatorname{int}\left(\frac{r_i}{\sum_{i=1}^{0.9N} r_i}\times 0.9\times N+0.5\right) \tag{3}$$

where int $(\cdot)$ stands for integer function.

1.4 "High quality" immigration

In order to avoid super individual and closed competition, individual difference is taken into consideration in deciding the necessity and quantity of immigration. The individuals of high quality whose fitness is greater than or equal to average fitness are required in immigration. When GH of the individuals in the mating set is less than λ, randomly produced high quality individuals immigrate in and randomly replace certain individuals till the average GH is greater than or equal to λ. λ is a constant, in most cases $\lambda = 0.01$.

1.5 Self-adaptive crossover operator

Crossover operates on $0.9N$ individuals in the mating set. For the two randomly chosen mating individuals with small GH (which means the individuals are similar), the value of crossover is not obvious, so the operation should decrease or stop. This means the crossover probability p_c should be small or zero. While with a greater GH, there are more chances to generate new individuals. So p_c should be greater in order to improve searching performance.

In GSAGA, p_c is formulated as

$$p_{ijc}=\begin{cases}0, & \alpha+\dfrac{\mathrm{GH}_{ij}-\overline{\mathrm{GH}}}{0.001+\overline{\mathrm{GH}}}<0\\[2ex] \alpha+\dfrac{\mathrm{GH}_{ij}-\overline{\mathrm{GH}}}{0.001+\overline{\mathrm{GH}}}, & 0\le\alpha+\dfrac{\mathrm{GH}_{ij}-\overline{\mathrm{GH}}}{0.001+\overline{\mathrm{GH}}}\le 1\\[2ex] 1, & \alpha+\dfrac{\mathrm{GH}_{ij}-\overline{\mathrm{GH}}}{0.001+\overline{\mathrm{GH}}}>1\end{cases} \tag{4}$$

where i, j are two mating individuals; GH_{ij} is the GH between two mating individuals; $\overline{\mathrm{GH}}$ is the average GH of all the individuals in the mating set; α is a constant (0.2–0.8).

p_c is the variable along with the mating individual. The larger the distance, the more chances for crossover there are, and *vice versa.*

1.6 Self-adaptive mutation operator

Self-adaptive operator guarantees solving each problem on the solution space and achieving global convergence. A small $\overline{\mathrm{GH}}$ suggests individuals of the population are approaching unanimity, and the schema of the population is monotonous. This might result in stagnation of evolution. The algorithms might converge at local optimal solution too early. Mutation operation can change this situation.

Before crossover operation, GH is used to evaluate the difference between the mating individuals. The individual difference, the individual fitness and $\overline{\mathrm{GH}}$ decide the mutation probability p_m.

In GSAGA, the individuals mutate according to the following p_m after crossover operation:

$$p_m=\frac{\beta}{(f_{\max}-\bar{f})\times\overline{\mathrm{GH}}+(\mathrm{GH}-\overline{\mathrm{GH}})+0.001} \tag{5}$$

where $f_{\max}$ is the maximum fitness, $\bar{f}$ is the average fitness, $\overline{\mathrm{GH}}$ is the average general hamming distance, and β is a constant, often as 0.005.

$f_{\max}$, $\bar{f}$ and $\overline{\mathrm{GH}}$ reflect the overall convergence degree of the population. When they are small, which means the population is converging, the p_m should be greater.

After immigration, crossover and mutation, $0.9N$ individuals in the mating set enter a new generation of population.

1.7 Condition for convergence

In GSAGA, if there is no conspicuous improvement of optimal fitness in M generations, the algorithm ends. When M is too large, it takes too much time to converge. While M is too small, there is a wide gap between the genuine optimal solution and the one having been obtained. Assume $M=N/3$ according to population size N.

2 Simulation of GSAGA

In reference [3], some modified genetic algorithms are compared with each other. The same optimization problem is used to evaluate GSAGA, and the results are compared with that of in reference [3].

2.1 Optimization problem

The object function

$$J=\frac{0.5-\sin^2(\sqrt{x^2+y^2})}{1+[0.001\times(x^2+y^2)]^2} \tag{6}$$

where $x\in[-100,+100]$, $y\in[-100,+100]$.

The optimization problem is to find the maximum value of J.

The function features a few maximums and they are very close to submaximums. The genuine optimal solution is 0.5, obtained at point (0, 0).

2:2 Realization of algorithms

The domain of variable x and y are converted to decimal strings of length 8, so that -100 is represented as "00000000" and $+100$ is represented as "99999999". The length of individual string, represented as k, is 16. Population size $N=3k=48$. Condition for stopping iteration: $M=N/3=16$.

The fitness function is defined as

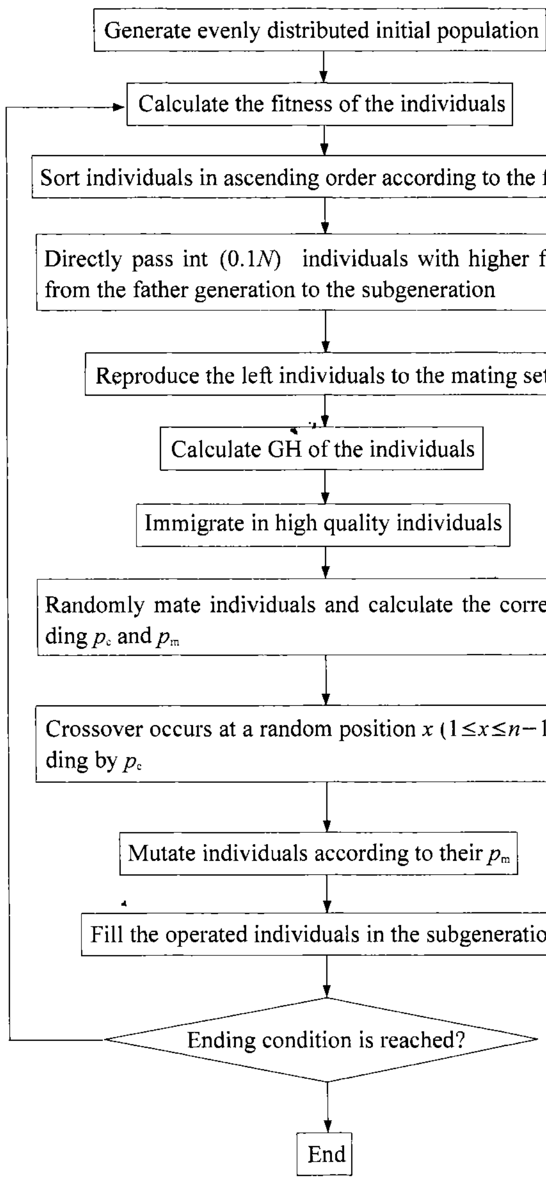
$$f_i=J_i-J_{\min}+\frac{J_{\max}-J_{\min}}{N} \tag{7}$$

where i represents the ith individual, $J_{\min}$ represents the minimum object function of the population, and $J_{\max}$ represents the maximum object function of the population.

A flow chart for the process is shown in **figure 1**, programmed in VB.

2.3 Result Analysis

In order to get rid of the disturbances caused by randomness, the algorithms are repeated 30 times. Experimental result data is shown in **table 1**. From the table, it shows that when the object function is less than



Generate evenly distributed initial population
Calculate the fitness of the individuals
Sort individuals in ascending order according to the fitness
Directly pass int (0.1N) individuals with higher fitness from the father generation to the subgeneration
Reproduce the left individuals to the mating set
Calculate GH of the individuals
Immigrate in high quality individuals
Randomly mate individuals and calculate the corresponding p_c and p_m
Crossover occurs at a random position x $(1\le x\le n-1)$ abiding by p_c
Mutate individuals according to their p_m
Fill the operated individuals in the subgeneration
Ending condition is reached?
End

Figure 1 GSAGA flow chart.

0.499 5, the algorithm converges to the given extreme soon. It also shows without immigration, the object function greater than 0.499 5 can be searched through many generations of evolution, or even fail to search the given extreme in certain generations.

Compared with the results of reference [3] (**table 2**) and many other modified GA, obvious advantages are shown in GSAGA; and the searching performance and convergence are greatly improved.

3 Conclusions

In GSAGA, the reasonable distribution of individual schema is guaranteed by generating initial population which is adaptive to solution space. Superior individuals are not broken because of crossover and mutation operation for they are sent to subgeneration directly. Meanwhile, high quality immigrants are introduced conditionally; and the diversity of individuals in the

Table 1 Generations for GASGA.

Runs of simulation	0.45	0.48	0.49	0.495	0.499	0.499 5	0.499 9	0.4999 5
1	6	6	9	9	13	13	19	9
2	3	3	5	5	15	15	15	29
3	8	8	8	8	9	16	16	211
4	4	10	10	11	43	48	49	63
5	1	1	22	28	56	107	163	166
6	7	13	47	47	88	167	167	248
7	0	9	12	45	45	110	144	144
8	7	7	8	8	17	43	44	44
9	4	4	4	8	159	211	286	287
10	0	0	7	7	13	13	13	119
11	0	9	9	9	42	85	120	181
12	13	17	45	45	60	60	60	243
13	5	7	7	7	10	10	10	93
14	8	9	11	11	11	46	46	52
15	3	4	9	9	40	102	155	156
16	17	18	46	46	81	96	120	123
17	7	10	56	56	56	56	149	206
18	0	0	31	31	31	94	167	167
19	2	2	2	2	2	2	2	2
20	2	2	2	9	9	20	23	50
21	7	7	10	10	10	10	10	97
22	4	11	11	12	12	16	16	180
23	5	7	7	9	22	22	285	285
24	0	17	17	17	17	17	21	30
25	0	7	7	7	7	10	21	72
26	8	12	12	21	23	23	48	620
27	7	10	10	10	14	14	19	111
28	3	3	3	14	14	14	21	146
29	11	14	14	21	27	29	34	130
30	13	18	50	50	75	106	106	348
average	5.17	8.17	16.37	19.07	34.03	52.50	78.30	154.73

Table 2 Generations for different algorithms in reference [3].

Object function	0.45	0.48	0.49	0.495	0.499	0.499 5	0.499 9	0.499 95
Simple genetic algorithm (SGA)	8	15	28	41	129	*	*	*
Uniform crossover	6	15	24	39	117	195	*	*
Non-biased immigration GA	7	16	27	38	*	*	*	*
GA in reference [3]	6	14	24	37	67	103	153	199

Note: * means the object function fails after the algorithm having run 200 generations.

population is provided efficiently. Finally, p_c and p_m of the individuals in the mating set are fixed self-adaptively, which solve the coordination problem between convergence and searching performance. In GSAGA, from initiation of population to immigration and the fix of p_c, p_m are all determined by the specific condition of the individuals, and are variable along with evolution. GSAGA, which embodies excellent searching performance and convergence quality, is superior to many existing GA.

References

[1] John H. Holland: *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.

[2] David E. Goldberg: *Genetic Algorithms on Search, Optimization and Machine Learning*. Addisen-Wesley, 1989.

[3] Hong Cai, Yanda Li: [in] *Proceedings of The Second Chinese World Congress on Intelligent Control and Intelligent Automation*. Xi'an Jiaotong University Press, Xi'an, 1997, p. 1689-1692.