Simultaneously Applying Multiple Crossover and Mutation Operators

Tzung-Pei Hong
Information Management Dept.
I-Shou University
Kaohsiung, 84008, Taiwan, R.O.C.

Hong-Shung Wang
Institute of Electrical Engineering
Chung-Hua University
Hsin-Chu, 30067, Taiwan, R. O. C.

Wei-Chou Chen
Information Management Dept.
I-Shou University
Kaohsiung, 84008, Taiwan, R.O.C.

1 INTRODUCTION

Traditional genetic algorithms use only one crossover and one mutation operator to generate the next generation. Determining which crossover and mutation operators should be used is quite difficult and is usually done by try-and-error. In this paper, a new genetic algorithm, the dynamic genetic algorithm, is proposed to solve the problem. The dynamic genetic algorithm simultaneously uses more than one crossover and mutation operators to generate the next generation. The crossover and mutation ratios changes along with the evaluation results of the respective offspring in the next generation.

2 DYNAMIC GENETIC ALGORITHM

Step 1: Define a suitable representation for the problem to be solved.

Step 2: Create an initial population set of ***N*** individuals for evolution.

Step 3: Define a suitable fitness function to evaluate the individuals.

Step 4: Form a mating set ***M***. The parents are then picked up at random from the mating set ***M***.

Step 5: Assign several appropriate or common candidate crossover and mutation operators, each with an initial ratio.

Step 6: Apply the crossover operators to generate the offspring (each crossover operator uses its ratio).

Step 7: Evaluate the fitness value of each individual and calculate the average growth value generated by each crossover operator. Assume parents *p* and *q* are chosen by a crossover operation to produce two offsprings *a* and *b*. The growth value is:

(The sum of the first two biggest fitness values among f(p), f(q), f(a), f(b)) - (f(p)+f(q)).

Step 8: Adjust the crossover ratios of the candidate crossover operators according to the average growth values.

Step 9: Apply the mutation operators to generate the offspring.

Step 10: Evaluate the fitness value of each individual and calculate the average growth value generated by each mutation operator. Assume a new string *p* is chosen by a mutation operator to produce a new offspring *a*. The progress value of the mutation operator is calculated as follows:

(The bigger fitness value between f(p) and f(a)) - f(p).

Step 11: Adjust the mutation ratios of the candidate mutation operators according to the average growth values.

Step 12: Select the superior *N* individuals according to the fitness values.

Step 13: If the termination criteria are not satisfied, go to Step 4; otherwise, stop the algorithm.

3 EXPERIMENTAL RESULTS

Experiments were made for comparing the performance of the proposed dynamic genetic algorithm with that of DCGA (dynamic crossover operators) and DMGA (dynamic mutation operators) using the following funciton:

Find the maxima:

$$f = x^4 \,|\sin(\pi x)\,|, 0 < x < 16.$$

Results are shown in Figure 1.

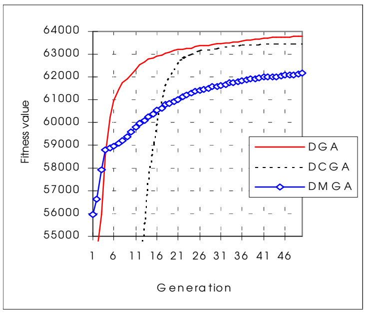

Figure 1: Experimental results

From the experimental results, we can find that DGA is better than DCGA and DMGA. DGA combines the advantages of these two methods and therefore converges faster and gets better fitness values than DCGA and DMGA.