Performance Improvement of Genetic Algorithms by Strong Exploration and Strong Exploitation

Sung Hoon Jung[1]

[1] Department of Information & Communication Engineering, Hansung Univ.
E-mail: shjung@hansung.ac.kr

Abstract

A new evolution method for strong exploration and strong exploitation termed queen-bee and mutant-bee evolution is proposed based on the previous queen-bee evolution [1]. Even though the queen-bee evolution has shown very good performances, two parameters for strong mutation are added to the genetic algorithms. This makes the application of genetic algorithms with queen-bee evolution difficult because the values of the two parameters are empirically decided by a trial-and-error method without a systematic method. The queen-bee and mutant-bee evolution has no this problem because it does not need additional parameters for strong mutation. Experimental results with typical problems showed that the queen-bee and mutant-bee evolution produced nearly similar results to the best ones of queen-bee evolution even though it didn't need to select proper values of additional parameters.

Key Words : Genetic Algorithms, Exploration, Exploitation, Optimization

1. Introduction

The exploration and exploitation in genetic algorithms are very important factors for improving the performances of genetic algorithms [2,3]. From this viewpoint, we have introduced queen-bee evolution for fast evolution of individuals by employing strong exploitation and strong exploration [1].

This evolution improved the performances of genetic algorithms about 200 times to 1,000 times than the simple genetic algorithms. However, it needs two additional parameters for strong mutation in order to control the exploration. This is a very critical drawback of the method because the two parameters ranged from 0 to 1 greatly affects the performances of genetic algorithms and there is no systematic method to decide proper values. Currently, only trial-and-error method, which is a very ineffective and time-consuming task, can be used for selecting proper values.

In this paper, we propose a new evolution method called queen-bee and mutant-bee evolution in order to overcome this problem. The proposed method does not need the two parameters for strong mutation no more. In the queen-bee and mutant-bee evolution, we adopt mutant-bees, the strongly mutated individuals. These mutant-bees for strong exploration are recombined with the queen-bees for strong exploitation to generate offsprings.

In the previous queen-bee evolution, the queen-bee, the fittest individual, is recombined with the normal individuals and the offsprings are strongly mutated by the two parameters, strong mutation rate and strong mutation probability. However, the normal individuals in the queen-bee and mutant-bee evolution are first strongly mutated and then the strongly mutated individuals (mutant-bees) are recombined with the queen-bee without additional parameters.

The proposed method has been tested with one combinational problem and two typical function optimization problems that are same as those of [1]. It was shown from the experiments that the performances of the queen-bee and mutant-bee evolution were

very similar to the best performances of the queen-bee evolution.

2. Queen-bee and Mutant-bee Evolution

In order to certainly compare the queen-bee and mutant-bee evolution and the queen-bee evolution, we describe them in Algorithm 1 and in Algorithm 2, respectively. The newly added or modified operations to the simple genetic algorithm are marked by ■ for the queen-bee and mutant-bee evolution and ▲ for the queen-bee evolution, respectively. In both evolution methods, from the first to the selection of parents to generate offsprings are same as shown in the Algorithms 1 and 2.

Parents are composed of the pair of queen-bee $I_q(t-1)$ and selected individuals $I_m(t-1)$ by a selection method such as roulette wheel selection and rank selection. In queen-bee evolution, those parents are recombined and then strongly mutated with the strong mutation rate ξ and strong mutation probability p_m'. On the other hand, in queen-bee and mutant-bee evolution the individuals $I_m(t-1)$ are strongly mutated by inverting their half most significant bits of strings before recombination. For example, the 8 bits of an individual, 01100111, is inverted to 10010111.

The strongly mutated individuals $I_m(t-1)^*$ (strong exploration) are recombined with the queen-bee (strong exploitation) to generate offsprings. Even though the queen-bee and mutant-bee evolution as described above is very simple compared to the queen-bee evolution and has no additional parameters, it shows nearly same performances to the best results of queen-bee evolution as shown in next section.

Algorithm 1 queen-bee and mutant-bee evolution
// t : time
// n : population size
// P : populations
// I_q : a queen-bee individual
// I_m : normal individuals
// I_m^* : inverted individuals (mutant-bees)
1 t = 0
2 initialize $P(t)$
3 evaluate $P(t)$
4 **while** (**not** termination-condition)
5 **do**
6 t = t + 1
7 select $P(t)$ from $P(t-1)$ (■)
8 $P(t) = \{(I_q(t-1), I_m(t-1))\}$
9 make mutant-bee (■)
10 invert $I_m(t-1)$ to $I_m(t-1)^*$
11 set $P(t) = \{I_q(t-1), I_m(t-1)^*\}$
12 recombine $P(t)$
13 do crossover
14 do mutation
15 evaluate $P(t)$
16 **end**

Algorithm 2 Queen-bee evolution
// t : time
// n : population size
// P : populations
// ξ : normal mutation rate
// p_m : normal mutation probability
// p_m' : strong mutation probability
// I_q : a queen-bee individual
// I_m : normal individuals
1 t = 0
2 initialize $P(t)$
3 evaluate $P(t)$
4 **while** (**not** termination-condition)
5 **do**
6 t = t + 1
7 select $P(t)$ from $P(t-1)$ (▲)
8 $P(t) = \{(I_q(t-1), I_m(t-1))\}$
9 recombine $P(t)$
10 do crossover
11 do mutation (▲)
12 **for** $i = 1$ **to** n
13 **if** $i \le (\xi \times n)$
14 mutation with p_m
15 **else**
16 do mutation with p_m'
17 **end if**
18 **end for**
19 evaluate $P(t)$
20 **end**

3. Experimental Results

The proposed method was tested on one combinational problem and two function

optimization problems used at [1].

Table 1. Experimental results

functions			f_1		f_2		f_3	
NE			64201.3	40495.1	59078.5	55933.1	776085.3	624688.8
QBE [1]	ξ	p_m	avg.	dev.	avg.	dev.	avg.	dev.
	0.4	0.6	40039.6	30811.1	50147.8	60182.2	944407.3	627457.7
		0.8	56686.6	40370.3	24768.4	15588.8	613873.5	534078.5
		1.0	6933.1	5944.4	10347.2	8666.2	7701.1	7366.8
	0.6	0.6	1115.3	695.8	1560.8	1486.0	355116.1	309387.3
		0.8	654.9	511.8	2354.5	2428.4	295721.0	261750.9
		1.0	351.0	392.3	962.6	893.7	4170.3	3277.9
	0.8	0.6	84.2	51.6	732.4	657.4	72121.8	70986.7
		0.8	97.5	59.6	349.2	259.6	37867.0	26954.4
		1.0	77.5	53.2	(*)298.0	199.8	(*)3767.3	2191.4
	1.0	-	(*)58.4	19.9	1840432.7	2234964.3	17779.8	16727.3
	best		58.4	19.9	298.0	199.8	3767.3	2191.4
QBMBE			89.7	80.7	204.8	172.6	6630.1	7145.5

$$f_1 = \sum_{j=1}^{h} m_j \quad \begin{cases} m_j = 1 & \text{if } T_j = I_j \\ m_j = 0 & \text{if } T_j \neq I_j \end{cases}$$

$$f_2 = 100(x_1^2 - x_2)^2 + (1 - x_1)^2, \quad where -2.048 \le x_i \le 2.048 \qquad (1)$$

$$f_3 = 0.5 - \frac{\sin(\sqrt{x_1^2 + x_2^2})\sin(\sqrt{x_1^2 x_2^2}) - 0.5}{(1.0 + 0.001(x_1^2 + x_2^2))(1.0 + 0.001(x_1^2 + x_2^2))}, \quad where -10 \le x_i \le -10$$

In equation (1), f_1 is a bit pattern matching problem between the target pattern T and an individual's pattern I; and f_2 and f_3 are DeJong function 2 and Mexican hat function, respectively. The parameters of genetic algorithms for experiments are as follows: the crossover probability p_c is 0.6; the mutation probability p_m is 0.05; the population size is 10; the individual length is 24 bits. Experimental results are shown in Table 1.

All results in the Table 1 are average values of 10 runs with different random number seeds. In Table 1, avg., dev., NE, QBE, and QBMBE mean average values of 10 runs, standard deviation of 10 runs, normal evolution. the queen-bee evolution [1], and the queen-bee and mutant-bee evolution, respectively. Note that the QBMBE outperforms NE of original genetic algorithm in all functions and shows similar performances to the best of QBE.

Since selected individuals as parents in queen-bee evolution are first recombined and then strongly mutated, the offsprings are mainly affected by the strong mutation. This makes the queen-bee evolution dependent on the strong mutation rate and strong mutation probability. However, the queen-bee and mutant-bee evolution shows relatively stable and good performances because its strong mutation method is more simple than the queen-bee evolution and generated offsprings are not affected by the strong mutation unlike the queen-bee evolution.

4. Conclusion

In this paper, we proposed a new evolution method, queen-bee and mutant-bee evolution. It was found from experiments that the proposed evolution showed very similar results to the best ones of the queen-bee evolution even though the proposed evolution does not need selecting of proper values of additional two parameters by a trial-and-error method. This indicates that the proposed method can be largely applied to existing genetic algorithms for improving their performances with simple modification and without additional efforts.

References

[1] S. H. Jung, "Queen-bee evolution for genetic algorithms," *Electronics Letters*, vol. 39, pp. 575-576, Mar. 2003.

[2] J. A. Vasconcelos, J. A. Ramirez, R. H. C. Takahashi, and R.˜R. Saldanha, "Improvements in Genetic Algorithms," *IEEE Transactions on Magnetics*, vol. 37, pp. 3414-3417, Sept. 2001.

[3] E. Alba and B. Dorronsoro, "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 9, pp. 126-142, Apr. 2005.