

# Fast Evolution by Multiple Offspring Competition for Genetic Algorithms

Sung Hoon Jung<sup>1</sup>

Department of Information and Communication Engineering, Hansung University, Seoul 136-792, Korea

## Abstract

The premature convergence of genetic algorithms (GAs) is the most major factor of slow evolution of GAs. In this paper we propose a novel method to solve this problem through competition of multiple offspring of individuals. Unlike existing methods, each parents in our method generates multiple offspring and then generated multiple offspring compete each other, finally winner offspring become to real offspring. From this multiple offspring competition, our GA rarely falls into the premature convergence and easily gets out of the local optimum areas without negative effects. This makes our GA fast evolve to the global optimum. Experimental results with four function optimization problems showed that our method was superior to the original GA and had similar performances to the best ones of queen-bee GA with best parameters.

**Key Words :** Genetic algorithms, premature convergence problem, function optimization, multiple offspring competition

## 1. Introduction

Although genetic algorithms (GAs) have been widely used to many engineering problems, the premature convergence problem of GAs was always an obstacle of successful applications because it caused slow evolution of GAs [1–4]. This premature convergence is that relatively good initial individuals located at local optimum areas become dominant after some generations because the selection pressure of GAs is high. If we control the selection pressure to low, then the evolution speed to the global optimum is also decreased. Therefore, this is not the solution. Once a GA falls into premature convergence, it is very difficult for individuals to get out of the local optimum areas because most individuals located small local optimum areas are similar to each other, so generated offspring are also similar to their parents. As a result, individuals stay long time at the local optimum areas. This is so called the premature convergence problem of GAs.

In order to solve this problem, a lot of methods have been introduced to date [4–7]. However, they didn't provide a fundamental solution in that their methods had negative effects even if they showed relatively good performances under appropriate parameters. This is because preventing from falling the premature convergence also causes slow evolution of GAs and forcing fast evolution also makes GAs fall into premature convergence. In other words, keeping the diversity of individuals in order not to fall into premature convergence also enables GAs not to

converge to the global optimum. As an example, if we increase the mutation probability to keep the diversity of individuals, then it also destroys the good building blocks of good individuals and results in slow evolution of GAs. This is a dilemma.

In this paper, we propose a simple, but effective and fundamental solution by employing competition of multiple offspring. In the reproduction process of GAs, our method unlike the existing methods generates multiple offspring and then generated multiple offsprings compete each other, finally winner offspring become to real offspring. For generating multiple offspring, we used three generation strategies: normal generation [1], strongly mutated generation, and queen-bee generation [6]. At the initial stage of GA, the offspring normally generated may be mainly winner because the individuals of GA have diversity and sometimes the offspring generated by queen-bee generation may accelerate the speed of evolution. If the individuals of GA fall into local optimum areas, then the offspring generated by strongly mutated generation may be often winner and this helps the GA get out of the local optimum areas.

Our method is a fundamental solution of the premature convergence problem of GAs because our method unlike existing methods has no negative effects. That is, if the good building blocks of good individuals are destroyed by the strong mutation for keeping diversity, then the offspring will not be winner and so it does not affect to the next generations. From these properties, our GA rarely falls into the

---

Manuscript received Sep. 22, 2010; revised . Nov. 20, 2010;  
accepted Dec. 2, 2010

Corresponding Author : shjung@hansung.ac.kr (Sung Hoon Jung)

A preliminary version of this paper was presented at the 25th International Technical Conference on Circuits/Systems, Computers and Communications, Thailand (July 2010). This research was financially supported by Hansung University.

premature convergence and easily gets out of the local optimum areas without negative effects. This makes our GA fast evolve to the global optimum without falling in local optimum.

In order to measure the performances of our method, we experimented our GA with four function optimization problems and compared its results to those of original GA [1] and those of queen-bee GA [6]. Experimental results showed that our GA was superior to the original GA and had similar results to the best ones of queen-bee GA with best parameters empirically selected. Although our method does not show better results than the queen-bee GA, our method is very effective than the queen-bee GA in that our method does not need additional parameters and empirical selection of the parameters.

This paper is organized as follows. Section 2 describes proposed multiple offspring competition method for genetic algorithms. Experimental results and discussion are provided in section 3. We conclude our paper in section 4.

## 2. Multiple Offspring Competition Method

This section first introduces the premature convergence problem and then describes proposed genetic algorithm with multiple offspring competition.

### 2.1 Premature Convergence Problem

Premature convergence addresses that relatively good individuals within local optimum areas at initial generations are repeatedly regenerated with small changes [5, 8, 9]. This makes most individuals be located within the local optimum areas after some generations. Once a GA falls into this local optimum areas, it is very hard for the individuals to get out of these areas because most individuals are similar to each other. In order to get out of the local optimum areas, GA should generate offsprings outside the local optimum areas (in other words, GA should increase the diversity of individuals). However, it is not easy because the crossover operation with similar parents cannot produce quite different offsprings from their parents. The mutation operation also cannot increase the diversity of individuals owing to the low mutation probability. As a result, individuals stay long time at the local optimum areas. In other words, it causes slow evolution of GAs.

In order to solve this premature convergence problem, a lot of works have been introduced to date [2–4, 10–13]. Their methods can be classified into three categories: deterministic methods, adaptive methods, and self-adaptive methods. Most existing methods regardless of categories have used the mutation probability as a control parameter of diversity. That is, they increased the mutation probability for increasing the diversity and vice versa. These con-

trol methods of mutation probability, however, had a negative effect that increasing the mutation probability could also increase the probability of destroying the good building blocks of good individuals. This causes another problem.

In this paper, we employed the multiple offspring competition in a reproduction process of GA. Unlike the previous methods, each parents in our method generate three types of offsprings according to the generation strategies, they compete with each other, and finally the winner among three offsprings become to real offspring. By adopting normal generation [1], strongly mutated generation, and queen-bee generation [6], our method operates appropriately for the situation of GA. That is, normal generation under the individuals have diversity will strongly affect to the individuals of the next generation while strongly mutated generation under most individuals fall into local optimum areas will increase the diversity of individuals. The queen-bee generation will accelerates the evolution of GAs without worrying about the premature convergence problem. Even if inappropriate offsprings are generated by some generation strategies, it will not affect to the individuals of next generation because they will not be winner. Finally, our multiple offspring competition method causes fast evolution of GAs without negative effects.

### 2.2 Proposed Algorithm

Our genetic algorithm with multiple offspring competition is described in Algorithm 1. The asterisks in Algorithm 1 mean additional operations to the original GA. Each parents as shown in the algorithm generates three offspring and they compete each other, finally the winner offspring become to real offspring for the parents.

---

**Algorithm 1** Genetic algorithm with multiple offspring competition

```

// t : time //
// P : populations //
// p1, p2 : selected individuals as parents //
// pq : queen-bee (the best) individual in a generation //
// pm, pm' : normal and strong mutation probability //
// o, o', oq : normal, strongly mutated, and queen-bee offspring //
// fo, fo', foq : fitness of offspring o, o', oq //
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   recombine P(t)
9   normal generation of offspring
10  do crossover with parents p1, p2
11  do mutation with pm

```

```

12     evaluate offspring  $o$ 
13     strongly mutated generation of offspring (*)
14     do crossover with parents  $p_1, p_2$  (*)
15     do mutation with  $p'_m$  (*)
16     evaluate offspring  $o$  (*)
17     queen-bee generation of offspring (*)
18     do crossover with parents  $p_1, p_q$  (*)
19     do mutation with  $p_m$  (*)
20     evaluate offspring  $o^q$  (*)
21     compete offsprings with  $f_o, f_{o'}$ , and  $f_{o^q}$  (*)
22     set the winner offspring to next offspring (*)
23     evaluate  $P(t)$ 
24 end

```

The normal generation is the same as that of original GA. In strongly mutated generation, crossover operation is the same as that of the original GA, but mutation operation is performed with strong mutation probability  $p'_m \gg p_m$ . From this, the strongly mutated offspring is quite different from their parents. If this strongly mutated offspring is better than the other offsprings, then it will be real offspring and helps the GA to keep diversity of individuals. That is, this enables GA not to fall into local optimum or to get out of the local optimum. Even if the strongly mutated offspring is quite worse than the other offsprings, it is not harmful to GA because it will not be real offspring. This is one of advantages of our method. In queen-bee generation, the queen-bee individual (in other words, the best individual in a generation) is always selected as the one of parents for all parents [6]. Since the queen-bee generation uses normal mutation probability, the offspring has many parts of good building blocks of the queen-bee individual. This sometimes makes it possible for the GA to fast evolve to the global optimum.

Since winner offspring become to real offspring, only successful generation strategy affects to the next generation. If a GA does not fall into local optimum areas, we can expect that normal generation plays a role of steady converge to the global optimum and the queen-bee generation often accelerates the speed of the convergence and strongly mutated generation sometimes finds new good areas. Otherwise, we can expect that the strongly mutated generation plays a role of increasing the diversity of individuals and sometimes normal generation and queen-bee generation complements the role.

### 3. Experimental Results

Our GA was tested on four function optimization problems as shown in Equation 1. Functions  $f_1$  to  $f_4$  are DeJong function 2, DeJong function 5, Mexican hat function, and Shafer function 2, respectively. Figure 1 shows the input-output relations of four functions. Function  $f_1$  is relative simple unlike the other functions in that it has only

one local optimum at the  $(-2.048, -2.048)$  point. Since the local optimum area, however, is quite broad, if the GA falls this area, then it is hard for the GA to get out of this area. Function  $f_2$  has many local optimum areas near only one optimum area. In the Mexican hat function  $f_3$ , since a lot of local optimum areas around only one global optimum at  $(0, 0)$  point exist, individuals rarely approach to the global optimum. Function  $f_4$  has four local optimum near  $(-10, -10)$ ,  $(10, 10)$ ,  $(10, -10)$ , and  $(-10, 10)$  at the four corners. These functions have been used so many tests because they have different properties.

We experimented with typical parameters as shown in Table 1. If a GA finds the global optimum, then the generation number is recorded. Since the performances of GA depend on the initial individuals, we experimented 10 runs with different random number seeds and averaged the results. Table 2 shows experimental results (we omit the standard deviation values for simplicity). In the table, OGA, QGA, and PGA mean original GA [1], queen-bee GA [6], and proposed GA, respectively. For the queen-bee GA, we set the parameters that showed best performances in the experiments [6], that is,  $\xi$  is set to 0.8 and the strong mutation probability to 1.0. Although the queen-bee GA showed great performances, it had a big drawback that its performances depended on the value of  $\xi$  and the strong mutation probability and there are no systematic selection methods. Our method without such drawback showed similar performances to the best results of the queen-bee GA. Only in function  $f_1$ , the PGA is inferior to the QGA because the function  $f_1$  has only one simple local optimum.

Since our method is devised not to fall into premature convergence and to get out the local optimum areas, it is effective for the functions having many local optima. As seen in the Table 2, the performance ratio QGA/PGA in function  $f_2$  is about 1.0. This indicates that the PGA has nearly same performance to the QGA. In more complex and difficult functions  $f_3$  and  $f_4$  that have many local optimum areas, the PGA sometimes outperforms the OGA. This result showed that our method could fast evolve the individuals without negative effects falling into local optimum areas.

In order to show the effects of three generation strategies, we take a typical experiment of function  $f_4$  with 30 population size and 24 individual length, which is the best result of our method compared to the queen-bee GA as shown in Table 2. In each generation, we count the number of winner offsprings according to the three generation strategies. Figure 2 shows the winner count of each generation strategy within initial generations and average fitness for showing the effects. As shown in the figure, initially normal and queen-bee generations are dominant, but the strongly mutated generation gradually becomes dominant after 5 generation. Note that the average fitness is increased as the winner count of strongly mutated offsprings and the

$$\begin{aligned}
 f_1 &= 100(x_1^2 - x_2)^2 + (1 - x_1)^2, \text{ where } -2.048 \leq x_i \leq 2.048 \\
 f_2 &= 0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a[i][j])^6}, \text{ where } -65.536 \leq x_i \leq 65.536 \\
 f_3 &= 0.5 - \frac{\sin(\sqrt{x_1^2 + x_2^2})\sin(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1.0 + 0.001(x_1^2 + x_2^2))(1.0 + 0.001(x_1^2 + x_2^2))}, \text{ where } -10 \leq x_i \leq 10 \\
 f_4 &= (x_1^2 + x_2^2)^{0.25} \sin(50(x_1^2 + x_2^2)^{0.1} + 1)^2, \text{ where } -10 \leq x_i \leq 10
 \end{aligned}
 \tag{1}$$

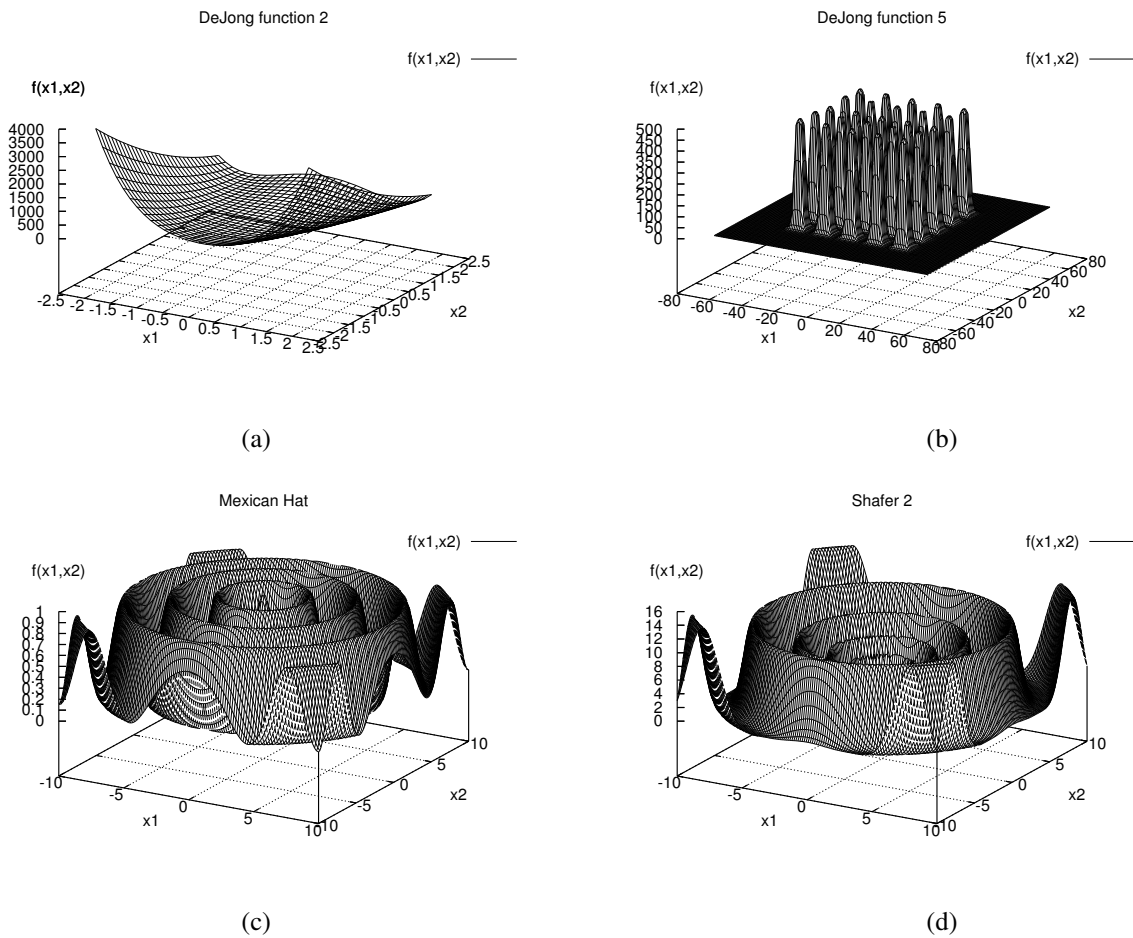


Fig. 1 Experimental functions (a)  $f_1$  (b)  $f_2$  (c)  $f_3$  (d)  $f_4$ )

Table 1. Parameters for experiments

Parameters	Values
Selection method	roulette wheel selection
Crossover probabilities	0.6
Normal and strong mutation probabilities ( $p_m, p'_m$ )	(0.05, 1)
Population size ( $p$ )	10, 20, 30
Individual length ( $l$ )	24, 28 bits

Table 2. Experimental results

fn	$l$	24				28			
		OGA	QGA	PGA	QGA/PGA	OGA	QGA	PGA	QGA/PGA
$f_1$	10	59078.5	208.4	18815.7	0.01	598051.8	359.0	12935.6	0.03
	20	17194.1	169.7	21254.0	0.008	174747.3	85.9	16736.5	0.005
	30	8377.9	80.0	14888.9	0.005	72868.3	57.0	28865.8	0.002
$f_2$	10	37401.3	657.8	910.9	0.7	1140458.0	3122.9	3857.9	0.8
	20	18759.6	132.3	134.8	0.99	257759.7	226.5	233.0	0.9
	30	13983.2	337.9	185.5	1.8	136731.8	142.3	165.2	0.86
$f_3$	10	776085.3	3570.2	1751.8	2.0	30914774.3	4098.6	2149.2	1.9
	20	521964.2	1719.1	1914.8	0.9	16675346.9	4433.3	1201.8	3.7
	30	414819.2	7085.8	1769.7	4.0	5490873.5	5971.2	2496.8	2.4
$f_4$	10	250111.6	12334.9	24335.5	0.5	3847880.3	67880.8	104062.9	0.7
	20	41878.4	47255.5	10622.2	4.4	900660.6	22126.0	38283.3	0.6
	30	41464.1	125109.1	8647.5	14.5	751085.6	147167.4	19991.1	7.4

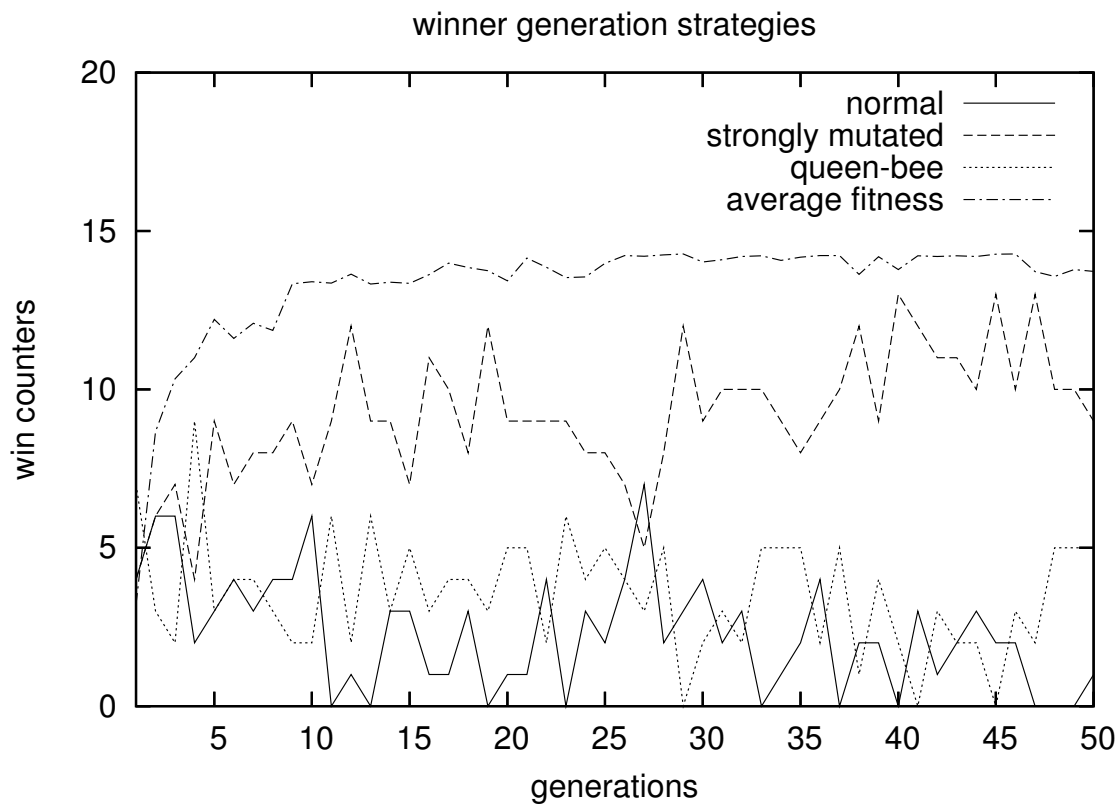


Fig. 2 Effects of three generation strategies

winner count of strongly mutated offsprings are nearly inverse to the winner count of normal offsprings. This result confirms our expectation.

#### 4. Conclusion

In this paper, we introduced a new method to solve the premature convergence problem of GA employing multiple offspring competition. Each parents generated three offspring using three generation strategies and winner offspring became to the real offspring. This allowed our GA to fast approach to the global optimum without falling into local optimum. It was found from experiments that our method showed very similar performances to the best results of the queen-bee GA and sometimes showed better performances than the queen-bee GA especially for complex functions having many local optimum areas. This means that our method is very effective and valuable.

#### References

- [1] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [2] S. M. Libelli and P. Alba, "Adaptive mutation in genetic algorithms," *Soft Computing*, vol. 4, pp. 76–80, 2000.
- [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.
- [4] A. E. Eiben, Z. Michalewicz, m. Schoenauer, and J. E. Smith, "Parameter Control in Evolutionary Algorithms," *Studies in Computational Intelligence*, vol. 54, pp. 19–46, 2007.
- [5] J. Andre, P. Siarry, and T. Dognon, "An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization," *Advances in engineering software*, vol. 32, no. 1, pp. 49–60, 2001.
- [6] S. H. Jung, "Queen-bee evolution for genetic algorithms," *Electronics Letters*, vol. 39, pp. 575–576, Mar. 2003.
- [7] S. Meyer-Nieberg and H.-G. Beyer, "Self-Adaptation in Evolutionary Algorithms," *Studies in Computational Intelligence*, vol. 54, pp. 47–75, 2007.
- [8] S. H. Jung, "Fast Evolution of Individuals through Competition of Multiple Offsprings in Genetic Algorithms," *The 25th International Technical Conference on Circuits/Systems, Computers and Communications*, July 2010. Thailand.
- [9] S. H. Jung, "Rank-based Control of Mutation Probability for Genetic Algorithms," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 10, pp. 146–151, May 2010.
- [10] L. Davis, "Adapting Operator Probabilities in Genetic Algorithms," in *Proceedings of the 3rd International Conference on Genetic Algorithms and their Applications*, pp. 61–69, 1989.
- [11] M. Srinivas and L. M. Patnaik, "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, pp. 656–667, Apr. 1994.
- [12] A. Tuson, "Adapting Operator Probabilities in Genetic Algorithms," master thesis, Dept. of Artificial Intelligence, University of Edinburgh, UK, 1995.
- [13] C. W. Ho, K. H. Lee, and K. S. Leung, "A Genetic Algorithm Based on Mutation and Crossover with Adaptive Probabilities," in *Proceedings of the 1999 Congress on Evolutionary Computation*, vol. 1, pp. 768–775, 1999.



**Sung Hoon Jung** received his B.S.E.E. degree from Hanyang University, Korea, in 1988 and M.S. and Ph.D. degrees from KAIST, in 1991 and 1995, respectively. He joined the Department of Information and Communication Engineering at the Hansung University in 1996, where he is a professor. His research interests are in the fields of intelligent systems, and in particular of application of neural networks, fuzzy logic, evolutionary computation algorithms to intelligent systems such as intelligent characters of computer games, computer generated forces on wargame, and so on. Recently, he started working on new research fields such as systems biology, bio-inspired engineering, brain engineering, and evolutionary complex systems. He is a member of the Korean Institute of Intelligent Systems (KIIS) and Institute of Electronics Engineers of Korea (IEEK).