Synergy of Multiple Crossover Operators in a Genetic Algorithm

Hyun-Sook Yoon and Byung-Ro Moon
School of Computer Science & Engineering, Seoul National University
Shillim-dong, Kwanak-gu, Seoul, 151-742 Korea
E-mail: hsy@soar.snu.ac.kr, moon@cs.snu.ac.kr

Recently there have been studies about using multiple operators in a genetic algorithm (GA) [1, 2]. It is important to determine appropriate operator probabilities in such a GA in order to achieve synergy of multiple operators. In this paper, we investigated various strategies in determining the operator probabilities in a GA with multiple crossover operators. We say that the crossovers have synergy if a combination of multiple crossovers performs better than the best one among them.

Given k different crossover operators $X_1, X_2, ..., X_k$, let C_i be the probability of applying X_i. Then, we can denote a combination of k different crossovers by $C_1X_1 + C_2X_2 + ... + C_kX_k$ (where $\sum_{i=1}^{k} C_i = 1$). In this paper, we study instances with $k = 2$ or 3. We used four strategies to determine operators' probabilities. Strategy 1 adaptively assigns an operator probability to each crossover according to the occupancy rate of the solutions generated by the crossover in the population. With a population of size N, let the number of the solutions that were generated by each crossover be n_i, $i = 1, 2, ..., k$ ($\sum_{i=1}^{k} n_i = N$). Then C_i becomes n_i/N. Strategy 2 is the opposite of Strategy 1. Strategy 3 maintains an occupancy rate for each crossover as close as possible to $1/k$. Strategy 4 maintains operator probabilities of all crossover operators with an expected rate $1/k$ regardless of the occupancy rates.

We tested with the traveling salesman problem (TSP) and used a steady-state hybrid GA with Lin-Kernighan algorithm. We chose three crossover operators (5-point, uniform, and cycle crossover) that are expected to afford significantly different search styles one another. We used TSP instances in the TSPLIB95[1] benchmark suite. We first measured the performance of each crossover separately. Next we chose two different crossovers out of the three and combined them. Three pairs of crossovers are possible. Lastly we combined all of the three crossovers. We examined the performance of the four strategies described above for all these combinations.

Table 1: Synergy-Effect Occurrences and Qualities

Graphs	1	2	3	4
rat575	1(1)	1(1)	2(2)	2(2)
att532	4(1)	3(1)	3(1)	4(1)
gr666	3(3)	0(0)	0(0)	1(1)
Total	8(5)	4(2)	5(3)	7(4)

Table 1 summarizes the experimental results. The Column "Graphs" has the instance names and the numbers in the title row indicate the strategies. Each element in the table is in the form "$x(y)$" where x is the number of synergy occurrences and y is the number of cases that the corresponding strategy performed best among the four strategies. Through the experiments, we could observe that the synergy effects depend much on the types of combinations, the strategies, and the instances. A notable phenomenon is that synergy effects usually came with running-time reduction. On the whole, Strategy 1 showed the strongest synergy and the other strategies also produced synergy effects in around 33% of the test cases.

In summary, combining multiple crossover operators in a GA did show synergy and relevant studies are widely open.

This work was supported by Brain Korea 21 Project.

[1] http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html

References

[1] I. Hong, A. B. Kahng, and B. R. Moon. Exploiting synergies of multiple crossovers: initial studies. In *IEEE Conference on Evolutionary Computation*, pages 245–250, 1995.

[2] W. M. Spears. Adapting crossover in evolutionary algorithms. In *the Fifth Annual Conference on Evolutionary Programming*, pages 367–384, 1995.