Mutation Rates in the Context of Hybrid Genetic Algorithms

Seung-Hee Bae and Byung-Ro Moon

School of Computer Science & Engineering, Seoul National University
Shilim-dong, Kwanak-gu, Seoul, 151-742 Korea
{baesoong, moon}@soar.snu.ac.kr

Abstract. Traditionally, the mutation rates of genetic algorithms are fixed or decrease over the generations. Although it seems to be reasonable for classical genetic algorithms, it may not be good for hybrid genetic algorithms. We try, in this paper, the opposite. In the context of hybrid genetic algorithms, we raise the mutation rate over the generations. The rationale behind this strategy is as follows: i) The perturbation rate of crossover decreases over the generations as the chromosomes in the population become similar; ii) Local optimization algorithms can undo a considerable level of perturbation and return the offspring to one of the parents; iii) Thus, we rather need stronger mutation at a later stage of a hybrid genetic algorithm. Experimental results supported our strategy.

1 New Mutation Strategy

There have been a great number of studies on the desirable mutation rates in Genetic Algorithms (GAs). The previous studies, however, were mostly applied to simple GAs that do not use local optimization heuristics. This study starts from the fact that hybrid GAs have significantly different characteristics from simple GAs. Because GAs are weak in fine-tuning around local optima, it is less probable to improve, by mutation, the qualities of high-quality solutions than of low-quality solutions. Thus it is a reasonable approach, as in the non-uniform mutation, to lower the mutation rates in later generations of GAs. In hybrid GAs, however, the population consists of the same or mostly similar local optimum solutions at a later stage. In the case, the perturbation by crossover is lower than in populations with enough diversity. If we apply weak mutation in this situation, the local optimization engine would return the offspring to one of the parents (local optima) highly probably. Operators in a simple GA should sometimes be able to produce better offspring than parents, whereas operators in a hybrid GA do not have to produce better offspring than parents because they only have to provide initial solutions for a local optimization heuristic. Thus operators are free from fine-tuning in a hybrid GA and this allows more perturbation of solutions than in a simple GA.

This work was supported by Brain Korea 21 Project. The ICT at Seoul National University provided research facilities for this study

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 381–382, 2004.

From this viewpoint, we suggest a new mutation strategy that perturbs gradually more over time. We incorporate the new mutation method in hybrid GAs with LK local optimization heuristic for the traveling salesman problem (TSP) and compare the experimental results against fixed and non-uniform mutations.

We devised two methods that changes the mutation rates. The first method increases the mutation rate if 50% of the population convergences. After increasing mutation rates, we keep the increased mutation rate until the population convergence rate drops below 50%. When the population convergence rate drops below 50%, it returns to the initial mutation rate. In the second method, we increase the mutation rate at every increase of 10% starting at the convergence rate 50%. The mutation rate is determined as $r = (1+k) \cdot r_0, k = max\{0, \lfloor 1 + \frac{c-50}{10} \rfloor\}$ where r is current mutation rate, r_0 is the initial mutation rate, and c is the current rate of convergence. In this case, it returns to the initial mutation rate whenever the best individual of the population is updated. We call the first strategy "Binary Increase" and the second one "Gradual Increase."

2 Experimental Results

Table 1 shows the experimental results of the four different mutation strategies on four TSPLIB* benchmark problems with 1000 to 4461 cities using 5-point crossover. The column "Type" represents the mutation type. "Fixed", "Non-Uni", "Bin-Inc", and "Grd-Inc" mean Fixed Rate, Non-Uniform, Binary Increase, and Gradual Increase, respectively.

Table 1. The Experimental Results

Instance	Type	Best	Avg(%)	GSD	Time(Gen)
dsj1000	Fixed	**18659688**	18659957.45(0.001)	53.19	1358(8468)
(18659688)	Non-Uni	**18659688**	18660082.53(0.002)	129.95	769(15000)
	Grd-Inc	**18659688**	**18659923.81(0.001)**	13.05	1519(8286)
	Bin-Inc	**18659688**	18659985.54(0.002)	96.99	1860(8230)
d2103	Fixed	**80450**	80461.94(0.015)	7.53	666(5410)
(80450)	Non-Uni	**80450**	80463.96(0.017)	5.73	530(10000)
	Grd-Inc	**80450**	**80457.33(0.009)**	5.89	1294(6623)
	Bin-Inc	**80450**	80463.19(0.016)	5.12	1467(6880)
pcb3038	Fixed	137698	137742.38(0.035)	5.06	1415(24873)
(137694)	Non-Uni	**137694**	137753.14(0.043)	5.70	658(30000)
	Grd-Inc	**137694**	137732.08(0.028)	2.63	1568(23658)
	Bin-Inc	**137694**	**137730.88(0.027)**	4.39	1755(22272)
fnl4461	Fixed	**182584**	182684.67(0.065)	8.32	1386(38404)
(182566)	Non-Uni	182605	182713.13(0.081)	8.25	1112(50000)
	Grd-Inc	182601	**182667.37(0.056)**	6.18	1878(45939)
	Bin-Inc	182606	182682.97(0.064)	8.70	2896(49907)

The proposed mutation strategies overall showed improvement over the Fixed Rate strategy. For the pcb3038 and fnl4461, the improvements were within 5% statistical risk. For the dsj1000 and d2103, the improvements were not within the stable range. Non-uniform mutation was inferior to the others.

* http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html