Single vs. Multiple Runs Under Constant Computation Cost *

Erick Cantú-Paz
Lawrence Livermore National Laboratory
cantupaz@llnl.gov

Suppose that we are given a fixed amount of function evaluations to solve a problem with a genetic algorithm (GA). How would we divide these evaluations to maximize the expected quality of the solution? One possibility would be use all the evaluations in one run of the GA with the largest population size possible. It is well known that the solution quality improves with larger populations. However, we could also use a smaller population and run the GA multiple times. The expected quality per run would decrease, but we would have more chances of reaching a good solution. The objective of this study is to determine what configuration reaches the best solutions. The scope of the results is limited to additively decomposable functions.

Some previous studies suggest that multiple runs are preferable, but it is not entirely clear under what conditions this holds, and there are conflicting reports. Also, if multiple populations were preferable, why are more practitioners not using them? This paper argues that multiple runs are preferable only in limited conditions, and that in most interesting cases the single largest run possible reaches the best solution. In addition, the results of this paper may explain some claims of superlinear speedups of multiple parallel runs and to determine when random search outperforms a GA.

The quality reached by multiple runs is better than one run if the following inequality holds:

$$\frac{\mu_{r:r}}{2\sqrt{m}} > P_1 - P_r. \qquad (1)$$

where P_r is the probability that each of the multiple runs reaches the desired solution, P_1 is the probability that the single largest possible run reaches the solution, m is the number of building blocks of the problem, and $\mu_{r:r}$ is the expected (normalized) quality of the best of r runs. This equation shows that multiple runs are more likely to be beneficial on short problems (small m), everything else being equal, but interesting problems may be very long. Multiple runs can also be advantageous when the difference between the solution qualities is small. This may happen at very small population sizes (where the quality is poor even in the single-run case) and when the quality does not improve much after a critical population size.

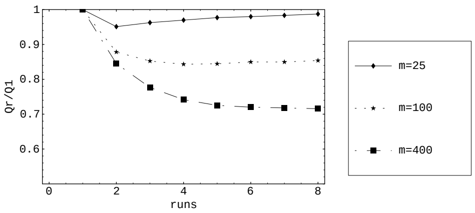

Figure 1: Ratio of the quality of multiple runs vs. a single run varying the size of the problem.

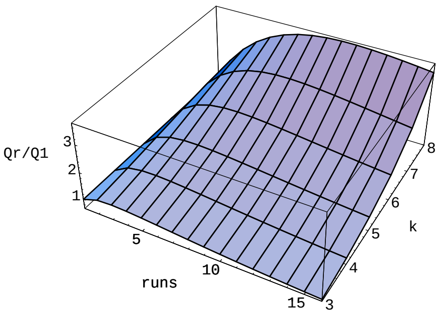

Figure 2: Quality of multiple runs of size 1 vs. a single run varying the order of the BBs and the number of runs.

Figure 1 shows that multiple runs perform better when m is small. We did experiments with the one-max function, keeping the population size per run fixed at $n_r = 10$. The results clearly show that as the problems become longer, the single large runs find better solutions than the multiple runs. Figure 2 shows predictions that show that random search (an extreme case of multiple runs) actually finds solutions that are better than a GA, expecially as the order of the BBs becomes larger. However, this only happens at low population sizes when $P_1 \approx P_r$. The results suggest that for difficult problems our best bet is to use a single run with the largest population possible.

* UCRL-JC-142172. This work was performed under the auspices of the U.S. Dept. of Energy by University of California LLNL under contract no. W-7405-Eng-48.