

## Genetic Algorithm Parameter Analysis

Ernesto BELMONT-MORENO

*Instituto de Física, UNAM, Apartado Postal 20-464, 01000 México D.F., México*

(Received October 11, 1999)

The energy minimizing problem of atomic cluster configuration and the 2D spin glass problem are used for testing our genetic algorithm. It is shown to be crucial to adjust the degree of mutation and the population size for the efficiency of the algorithm.

When a Genetic Algorithm (GA) is applied to solve any optimization problem, the questions arise; how to implement it; what structure to use; how much weight to give to the three main processes (mutation, recombination and reproduction).<sup>1)</sup> The present work is a part of a systematic study of the last question, in particular, what population size and, how much mutation? The procedure used is as follows: a standard GA is applied to two physical problems, a) the determination of the most stable configuration of an atomic aggregate, by minimizing the interaction energy<sup>2)</sup> and b) the calculation of the ground state of a two dimensional finite spin glass, also minimizing the energy.<sup>3)</sup> As the two problems are quite different (the spin glass is a combinatorial problem and the atomic problem is a real variable problem), a coincidence in some result is significant, so here only common conclusions are reported. The first result shows the clear existence of an optimal value for the mutation rate, see Figs. 1 and 2, the point of minimum energy in this graph indicates the best value for the mutation rate for the corresponding population size. Another observation is that, when increasing the population size the mutation continues being important

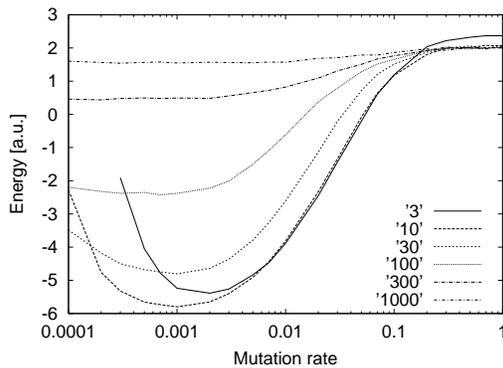


Fig. 1. Minimum energy as a function of mutation rate found for a cluster of 30 Na and 30 Cl atoms. The value shown is after 1000 energy evaluations averaging over 1000 different initial seeds. Each curve is for a different population size as indicated.

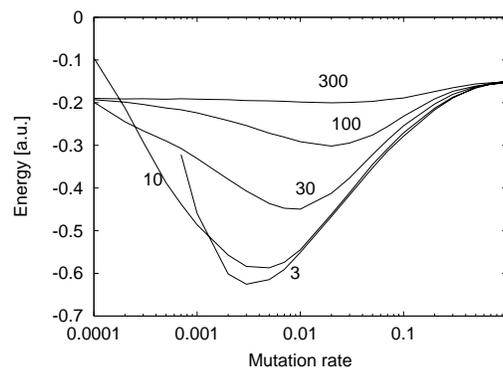


Fig. 2. Minimum energy as a function of mutation rate found for a spin glass of  $30 \times 30$ . The value shown is after 1000 energy evaluations averaging over 1000 seeds. Each curve is for a different population size as indicated.

unlike genetic programming<sup>4)</sup> and other evolutive algorithms. Observing that in the GA used the optimum mutation rate diminishes as the number of evaluations increase, we can conclude that it is good practice to decrease the mutation rate as a function of the number of evaluations. Also, as expected, the best mutation rate decreases with the problem size, see Fig. 3. Concerning the population size, from Figs. 1 and 2, for 1000 evaluations, it is clear that there exists an optimal value, around 10. Making the equivalent calculations for values from 10 to 10,000 evaluations, still the best size looks to be around 10 individuals. Even more, in the atomic example (180 variables), small change in the best population size is observed on going from 18 to 600 variables. As indicated previously, similar behavior is observed in both problems, compare Figs. 1 and 2. In conclusion, for a fixed calculation time (fixed number of evaluations) it is worthwhile to look for the appropriate best mutation and population size, the latter can be surprisingly small (more generations for a fixed number of evaluations). It is important to point out that the results found here are for the specific problems presented. Results obtained with GA are problem dependent.

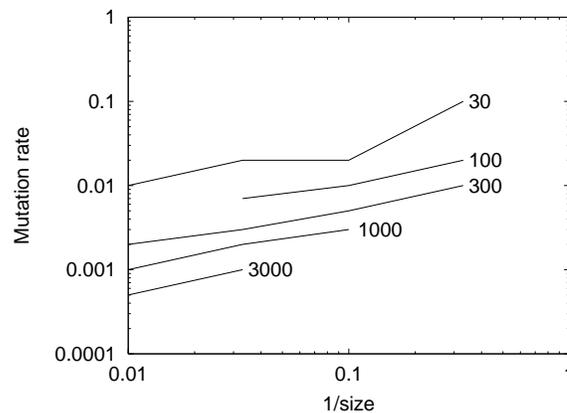


Fig. 3. Best mutation rate as function of the inverse of problem size. Each line corresponds to different number of evaluations. The problem is the same as Fig. 1 with the population fixed to 10 individuals.

### References

- 1) J. H. Holland, *Adaptation in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, 1975).
- 2) K. Michaelian, *Am. J. Phys.* **66** (1998), 231.
- 3) P. Sutton, D. L. Hunter and N. Jan, *J. Phys. I France* **4** (1994), 1281.
- 4) J. R. Koza, *Genetic Programming* (The MIT Press, Cambridge, Massachusetts, 1992).