Comparison of Simulated Annealing with Genetic Algorithms in Biological Problems that Use Recurrent Neural Nets

George Marnellos[1,2] and **Eric Mjolsness**[3]
1) Section of Neurobiology, Yale University, New Haven, CT
2) Sloan Center for Theoretical Neurobiology, The Salk Institute, La Jolla, CA
3) Department of Computer Science and Engineering, UCSD, La Jolla, CA
{gmarnell,emj}@cs.ucsd.edu

Abstract

We have used recurrent artificial neural nets (see Mjolsness *et al.*, 1991) to simulate how genes and their interactions in cells determine the phenotypes of animal organs or simple organisms and their development, as well as how such gene interactions evolve under particular (simulated) environmental conditions or constraints: nodes in these neural nets correspond to genes and node activation levels to gene expression levels. We optimize the parameters in these models (node interaction strengths, activation and decay rates, thresholds and so on) in order to either fit experimental data (gene expression patterns) or to impart desired features to the simulated system and make it conform to constraints.

We have examined stochastic optimization techniques for the analytically intractable energy functions of our models; we have compared their performance in terms of convergence speed and quality of solution. Specifically, we have used the techniques of simulated annealing (SA) and genetic algorithms (GA) on the following problems:

- Life history - simulating the development of a simple multicellular organism capable of reproducing and its evolution under various selective environments (see Mjolsness *et al.*, 1995),
- Neurogenesis - fitting gene-expression patterns observed during early neurogenesis in Drosophila, and
- Curve-fitting - fitting multi-dimensional Lissajous curves with the use of a fully-connected neural net (this being an application similar to but simpler than the above two, which incorporate multiple such nets).

The energy functions of the neurogenesis and curve-fitting problems are evaluated directly on node activation patterns, while in the life-history problem node activations are first mapped to phenotype traits of the organism and the energy function is then evaluated on those traits.

Both optimization methods we applied had several parameters that needed tuning. The SA schedule that we used has been described in Lam & Delosme (1988). Our GA is implemented in parallel (one subpopulation per processor); it consists of several constant-size subpopulations evolving in parallel (5 to 9 subpopulations each of 100 to 200 chromosomes) and has mutation, recombination and migration. It is an elitist GA, in that the best chromosome in each subpopulation always survives unchanged to the next generation. The fitness of only a fraction of mutated chromosomes is updated in each generation and this means that at any time there is a number of chromosomes with inaccurate fitness; in combination with low mutation and replacament rates, as we had in our runs, this reduces the number of evaluations of the energy function per generation to only a small fraction of the population size.

Our results indicate that the performance of the two optimization methods varies from problem to problem. In terms of total energy function evaluations, the GA is significantly faster on the life-history problem but SA is faster on the neurogenesis and curve-fitting problems (in terms of real time, the GA is comparable or slightly faster than SA on the curve-fitting problem, since the GA is implemented on several processors in paralel). SA often reaches solutions better by a factor of 2 or more, in terms of energy level, although that may require a very large number of energy function evaluations (and may take several days, in real time, on an SGI Indigo or an IBM PowerPC). It is not clear why the performance of the two optimization methods varies across apparently similar problems. One possible explanation is that energy functions evaluated directly on neural net node activations and energy functions that depend only indirectly on those activations may present different challenges to each of the two methods.

Acknowledgements

This work was partially supported by the Yale Institute for Biospheric Studies (Center for Computational Ecology), the Neuroengineering and Neuroscience Center at Yale and the Yale Center for Parallel Supercomputing. We also thank the GEURU group at the UCSD Computer Science Department for useful discussions.

References

Mjolsness, E., Sharp D.H., & Reinitz, J. (1991). A connectionist model of development. *J.Theor.Biol.*, 152, 429–453.

Mjolsness, E., Garrett, C.D., Reinitz, J. & Sharp, D.H. (1995). Modeling the connection between development and evolution: Preliminary report. In *Evolution and Biocomputation, Computational Models of Evolution* (pp. 103–122), Banzhaf, W. and Eeckman, F.H. (eds.). Springer, Berlin.

Lam, J. & Delosme, J.-M. (1988). An Efficient Simulated Annealing Schedule: Implementation and Evaluation. Technical Report 8817, Engineering Department, Yale University, New Haven, CT.