
Genetic Algorithm (GA) Tips

David L. Carroll

*CU Aerospace
2004 South Wright St. Extended
Urbana, Illinois 61802*

Welcome to my genetic algorithm tips World Wide Web page. Comments or questions should be sent to carroll@cuaerospace.com.

[Click here to return to my Fortran GA webpage.](#)

General Recommendations

Any users new to the GA world are encouraged to read David Goldberg's "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley, 1989.

I still recommend using the micro-GA technique (microga=1) with uniform crossover (iuniform=1). However, if possible, I strongly suggest that you use values of nposibl of 2^n (2, 4, 8, 16, 32, 64, etc.). While my test function works fine for other values of nposibl, I have encountered problems where the uniform crossover micro-GA has difficulty with parameters having long bit strings and a non- 2^n value of nposibl, e.g. nposibl=1000, will have 10 bits assigned (for this case I would suggest running nposibl=1024 rather than 1000); I am presently working on a fix for this situation.

For more conventional GA techniques I recommend using:

- Binary coding (only option with my GA)
 - Tournament selection (only option with my GA)
 - Uniform crossover (iuniform=1)
 - Creep mutations (icreep=1)
 - Niching or sharing (iniche=1)
 - Elitism (ielite=1)
-

micro-GA Technique

My favorite GA technique is still the micro-GA. At this point, I recommend using the micro-GA with uniform crossover and a small population size. The following inputs gave me excellent performance:

- microga = 1

- npopsiz = 5
- maxgen = 100
- iunifrm = 1

I have also gotten good performance with the single-point crossover (iunifrm=0), micro-GA.

If you decide to use the micro-GA, you will not need to worry about the population sizing or creep mutation tips below.

See the Krishnakumar reference below for more information about micro-GA's.

Population Sizing

I've had a lot of people ask me about population sizing, especially people who are attempting large problems where 100 individuals is probably not enough. The true authority on the subject is [David E. Goldberg](#) but here is a crude population scaling law in my paper (based on Goldberg & Deb, 1992):

- npopsiz = order[(l/k)(2**k)] for binary coding

where l = nchome and k is the average size of the schema of interest (effectively the average number of bits per parameter, i.e. approximately equal to nchome/nparam, rounded to the nearest integer). I find that when I have uniform crossover and niching turned on (which I recommend doing), that this scaling law is usually overkill, i.e. you can most likely get by with populations at least twice as small.

Remember to make the parameter 'indmax' (in 'params.f') greater than or equal to 'npopsiz'.

Creep Mutation Probability Tip

I generally like to have approximately the same number of creep mutations and jump mutations per generation. Using basic probabilistic arguments, it can be shown that you will get approximately the same number of creep and jump mutations when:

- pcreep = (nchome/nparam) * pmutate

where pmutate (the jump mutation probability) is 1/npopsiz.

Useful GA References

Goldberg, D. E., and Richardson, J., "Genetic Algorithms with Sharing for Multimodal Function Optimization," Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms, 1987, pp. 41-49.

Goldberg, D. E., "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley, 1989.

Goldberg, D. E., "A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing," in: Complex Systems, Vol. 4, Complex Systems Publications, Inc., 1990, pp. 445-460.

Goldberg, D. E., "Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking," in: Complex Systems, Vol. 5, Complex Systems Publications, Inc., 1991, pp. 139-167.

Goldberg, D. E., and Deb, K., "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms," in: Foundations of Genetic Algorithms, ed. by Rawlins, G.J.E., Morgan Kaufmann Publishers, San Mateo, CA, pp. 69-93, 1991.

Goldberg, D. E., Deb, K., and Clark, J. H., "Genetic Algorithms, Noise, and the Sizing of Populations," in: Complex Systems, Vol. 6, Complex Systems Pub., Inc., 1992, pp. 333-362.

Krishnakumar, K., "Micro-Genetic Algorithms for Stationary and Non-Stationary Function Optimization," SPIE: Intelligent Control and Adaptive Systems, Vol. 1196, Philadelphia, PA, 1989.

Syswerda, G., "Uniform Crossover in Genetic Algorithms," in: Proceedings of the Third International Conference on Genetic Algorithms, Schaffer, J. (Ed.), Morgan Kaufmann Publishers, Los Altos, CA, pp. 2-9, 1989.

If you are interested in my work (which may give some insights into how and why I coded some aspects of my GA), I can mail a copy of my three GA papers. I have also converted two of my GA papers into PDF files. You can read them directly via Adobe Acrobat Reader. If you do not already have this free software, follow this link to [download the latest version of Adobe Acrobat Reader](#).



G. Yang, L.E. Reinstein, S. Pai, Z. Xu, and D.L. Carroll, "A new genetic algorithm technique in optimization of permanent 125-I prostate implants," *Medical Physics*, Vol. 25, No. 12, December 1998, pp. 2308-2315.

- [MedPhys-1998.pdf](#) (7.4 MB)

Carroll, D. L., "Chemical Laser Modeling with Genetic Algorithms," *AIAA J.*, Vol. 34, 2, 1996, pp. 338-346.

- [AIAAJ-1996.pdf](#) (8.9 MB)

Carroll, D. L., "Genetic Algorithms and Optimizing Chemical Oxygen-Iodine Lasers," *Developments in Theoretical and Applied Mechanics, Vol. XVIII*, eds. H.B. Wilson, R.C. Batra, C.W. Bert, A.M.J. Davis, R.A. Schapery, D.S. Stewart, and F.F. Swinson, School of Engineering, The University of Alabama, 1996, pp.411-424.

- [SECTAM18.pdf](#) (0.2 MB)

For extensive Genetic Algorithm web-sites see:

- [Illinois Genetic Algorithm Laboratory](#)
- [The Navy Genetic Algorithm Archives](#)

[Click here to return to my Fortran GA webpage.](#)

[David L. Carroll](#)
carroll@cuaerospace.com