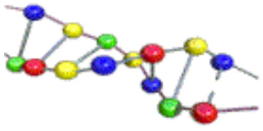
Introduction to
Genetic Algorithms



XIII. Recommendations

Parameters of GA

This chapter should give you some basic recommendations if you have decided to implement your genetic algorithm. These recommendations are very general. Probably you will want to experiment with your own GA for specific problem, because today there is no general theory which would describe parameters of GA for *any* problem.

Recommendations are often results of some empiric studies of GAs, which were often performed only on binary encoding.

- **Crossover rate**
  Crossover rate generally should be high, about **80%-95%**. (However some results show that for some problems crossover rate about 60% is the best.)
- **Mutation rate**
  On the other side, mutation rate should be very low. Best rates reported are about **0.5%-1%**.
- **Population size**
  It may be surprising, that very big population size usually does not improve performance of GA (in meaning of speed of finding solution). Good population size is about **20-30**, however sometimes sizes 50-100 are reported as best. Some research also shows, that best population size depends on encoding, on **size of encoded string**. It means, if you have chromosome with 32 bits, the population should be say 32, but surely two times more than the best population size for chromosome with 16 bits.

- **Selection**
  Basic **roulette wheel selection** can be used, but sometimes rank selection can be better. Check chapter about selection for advantages and disadvantages. There are also some more sophisticated method, which changes parameters of selection during run of GA. Basically they behaves like simulated annealing. But surely **elitism** should be used (if you do not use other method for saving the best found solution). You can also try steady state selection.
- **Encoding**
  Encoding **depends on the problem** and also on the size of instance of the problem. Check chapter about encoding for some suggestions or look to other resources.
- **Crossover and mutation type**
  Operators depend on encoding and on the problem. Check chapter about operators for some suggestions. You can also check other sites.

Applications of GA

Genetic algorithms has been used for difficult problems (such as NP-hard problems), for machine learning and also for evolving simple programs. They have been also used for some art, for evolving pictures and music.

Advantage of GAs is in their parallelism. GA is travelling in a search space with more individuals (and with genotype rather than phenotype) so they are less likely to get stuck in a local extreme like some other methods.

They are also easy to implement. Once you have some GA, you just have to write new chromosome (just one object) to solve another problem. With the same encoding you just change the fitness function and it is all.On the other hand, choosing encoding and fitness function can be difficult.

Disadvantage of GAs is in their computational time. They can be slower than some other methods. But with todays computers it is not so big problem.

To get an idea about problems solved by GA, here is a short list of some applications:

- Nonlinear dynamical systems - predicting, data analysis
- Designing neural networks, both architecture and weights
- Robot trajectory
- Evolving LISP programs (genetic programming)
- Strategy planning
- Finding shape of protein molecules
- TSP and sequence scheduling
- Functions for creating images

More information can be found through links in the appendix.

Previous Next
