

# Re: Minimization principal for evolution

---

*Source:* <http://sci.tech-archive.net/Archive/sci.bio.evolution/2006-02/msg00448.html>

---

- *From:* [anon1@xxxxxxx](mailto:anon1@xxxxxxx)
  - *Date:* Sun, 19 Feb 2006 00:49:14 -0500 (EST)
- 

There is a buzz word in computer science called "Genetic Algorithms", which try to use evolutionary principles to minimize objective functions.

The issue is really handicapped in two areas.

1) Evolution does not subscribe to exactly one objective function for all organisms, whereas all studies in optimization have one objective function by which all hypothesis are measured. Part of evolution is the development of different species for different niches (not the best species for one niche).

I think this is partly because most problems in computer science are single optimization problems (such as best route for salesmen, best electric circuit for pre-defined purpose, etc.) rather than arms-race problems, and partly because the field is just getting started and it takes time to develop software for arms-race problems compared to single-optimum problems. But a disadvantage when genetic algorithms are applied to non-arms-race problems is that the problem is in some sense irreducibly complex. The target is set way-up-high at the very start, and the algorithm must get toward that one target in order to show progress and thereby supply selection pressure. Natural evolution has it comparatively easier, because at the start the goal (defeating the enemy) is relatively easy to reach, and it's only after that goal has been nearly achieved that the goal starts to move, but it moves only a little bit at a time so it's easy for selection pressure to keep moving toward its new position, never being so far from the goal that the goal is invisibly far away.

Here's a problem type where genetic algorithms could make direct use of arms-race: Simulated terrorists opposed to anti-terrorism forces. Perhaps our Department of Homeland Security should give it a try.

2) There are much better algorithms for optimization, bootstrapping, monte carlo, gradient descent, etc... The genetic algorithms have problems with actually minimizing a function because one strand will dominate the gene pool, see issue number 1.

## Re: Minimization principal for evolution

I rather like the thermodynamic (annealing, water-level) approach. You use a genetic algorithm to do the actual optimizing, but you contrive a pretend arms-race by raising the water level thereby reducing the portion of land above water level every so often.

I'm not sure what you mean by "one strand". In a true genetic algorithm, you aren't evolving a single line of descent, you're running a whole population of genomes in parallel, and it's only when the count of some genome drops to zero that you stop emulating it with the others. Ideally you factor the phenotype somehow so that you can put different factors on different loci and emulate meiosis to mix-and-match the various loci/factors stochastically so that several factors and several combinations of factors can all evolve in parallel. One bad example is trying to optimize a computer algorithm expressed as a Von Neumann program, whereby the different instructions in a program module are so tightly intertwined that mix-and-match (cross-over) is fatal. But a production-rule algorithm might be more reasonably factored such that mix-and-match would work just fine. A stochastic production-rule algorithm, where competing rules would be picked at random, would work fine, and duplication events would work to increase the priority (chance of being picked) in case of conflict, so the emulation would be even more like natural evolution.

..

.