

CS369 Computational Science
Evolutionary Algorithms
Optional Exercise
Matthew Egbert 2017

Hands-on experience can really help you to remember what you have learned. For this reason I recommend that you gain some experience working with genetic algorithms by writing one of your own. This exercise is optional and not graded, but if you have questions along the way, feel free to email me or swing by my office.

Task #1. Write a genetic algorithm that maximises the sum of a set of N real-valued genes that are capped between (say) 0 and 1. You could start with the a minimal version of the microbial genetic algorithm and extend it with demes, or modify it to use alternative forms of sexual recombination rather than the microbial mechanism of overwriting the loser's genome with part of the parent's genome. This exercise is also a good place to investigate the different methods for limiting the allowed values of genes (clamping, wrapping and bouncing). In particular, what are the effects of wrapping?

Task #2. A different task is to evolve a solution to the following card problem.

You have 10 cards numbered from 1 to 10. You have to choose a way of dividing them into 2 piles, so that the cards in Pile_0 **sum** to a number as close as possible to 36, and the product of the remaining cards in Pile_1 is a number as close as possible to 360.

Why would no-body ever actually use a GA to solve this particular problem in real life?

Can you conduct any experiments that give you insight into the fitness landscape of this problem?