

Andrew Cleland
CS 541 Artificial Intelligence
Proposal for Coevolution Extra-Credit Project

Prof. Bryant York,

I propose to do a project implementing the spatial coevolution algorithm as presented in the MitchellWCCI2006 paper, applied to the density=1/2 cellular automata (CA) problem we studied.

The program will coevolve two populations: the “host” population, consisting of the CAs (defined by their update rules), and the “parasite” population, consisting of initial configurations (ICs).

Project Outline

- Host and parasite populations are spatially organized in a 2-dimensional grid, with periodic boundary conditions. Each point in the grid corresponds to a (CA, IC) pair.
- Fitness of each host is computed by running its CA on its own IC and all 8 neighboring ICs. The fraction correct is the CA’s fitness.
- Fitness of each parasite is computed only with respect to its paired CA. The fitness is 0 if the CA obtains a correct solution, and $|\text{density(IC)} - 1/2|$ otherwise. (As noted in the paper, this done to “limit the damage” of the parasite by prioritizing ICs that will be easier to solve).
- Host and Parasite Selection (quoting from MitchellWCCI2006):

For each host h , rank h along with the other eight hosts in its neighborhood according to fitness, with the highest fitness host having rank 1 and the lowest having rank 9. Each of these 9 hosts has probability of being selected equal to 0.5^{rank} (except for the bottom-ranking host, which has probability 0.5^8 , so that the nine probabilities will sum to 1). The selected host h' replaces the host h in the center site of this neighborhood (if h itself is selected, no change is made). The same selection procedure is applied to each parasite p , which competes similarly with the eight other parasites in its neighborhood. The replacement of hosts and parasites at each site are done synchronously.

- Crossover of CAs:
 - For each site in the grid:
 - If random number < crossover probability:
 - Choose a random neighbor to crossover with.
 - Crossover at a random point to produce two children
 - Select one child at random to replace the CA in the grid.
- Parasites do not crossover.
- Mutation of both CAs and ICs: all hosts and parasites are mutated by randomly choosing one or more bits and flipping their values.

- Repeat the above process for a set number of iterations, and observe improvements in CA performance, and look for cases of the “particle” strategy that Mitchell observed.

Programming Language:

I would most prefer to do this Python because it is the language with which I have the most facility (it also easier to do test components as I go along), but presuming that this not allowed, I will implement this in either C or Java.

Please let me know if you approve of this proposal and let me know of any comments or suggestions you have.

Thank you,

Andrew