Parallelizing a Simple Genetic Algorithm: Robbie the Robot

Robbie the Robot is a garbage-collecting robot who lives in a finite, 3×3 or larger gridworld strewn randomly with trash. Robbie sees in such low resolution that he knows only the state of the cells immediately to his north, south, east or west. Robbie must move to a cell to collect the garbage in that cell, and at every time step may either move to one of the cells he can see, attempt to collect garbage in his current cell, stay still, or move randomly. Robbie is penalized by the environment for trying to travel through walls as he cannot, and it damages his chassis. Robbie must develop a garbage collection strategy for each of the $3^4 - 13 = 68$ situations that he may encounter during his task (each visible cell may be either wall, filled with garbage, or empty, and the 13 situations where Robbie is adjacent to three or more walls do not occur in 3×3 or larger grids). This problem statement is due to Melanie Mitchell from *Complexity: A Guided Tour*.

The 68 (non-overlapping) bytes necessary to encode this strategy comprise Robbie's genome, which may be evolved in a number of ways; crossover and mutation are typical. The fitness of a genome is measured via simulation. There is no distinction between Robbie's genotype and phenotype.

There are several moving parts in the project. The core task of simulating the behavior of many genomes is embarassingly parallel without any modification. It is also necessary to have a graphical simulator; it is impossible to understand what a genome does otherwise. This is the most significant single-threaded component. A minimally-featured final product would consist of a graphical simulator, functions for parallel fitness evaluation of many genomes, fitness proportionate- or rank-selection, and parallel mutation and crossover at specified rates within a population along with some logging and graphing utilities.

In addition, I would like to extend the selection mechanism using parallelized versions of what we attempted in our third homework assignment. It is straightforward to imagine building a lexicographic-adjacency graph over genomes rather than words as it was useful to do for finding word ladders; traversals of this graph could then yield lots of useful information about which mutations and crossovers seem mostly likely to increase the fitness of offspring. The parallel construction, reading, and even modification of this graph could provide very interesting opportunities for parallelism, as well as an interesting twist on this genetic algorithm!

There are also plenty of ways to extend the task the genetic algorithm attempts to solve: one could modify the gridworld to contain obstacles or ”hidden” garbage (so that one of Robbie's actions is to also to check for garbage in his current cell); Robbie could be allowed to see larger areas; his genome could be represented by a decision tree or very simple neural network (both of which can also be evolved). These features are lower priority to me because they mostly do not invite new parallelism. Many of them also represent rather trivial modifications to fitness functions or to the data structures of the gridworld, Robbie's genome, or his state.

Any guidance on what kind of basic GUI tools I can expect to have available on the machine that will be used to evaluate this project are appreciated.

Thank you!

Eli Whitehouse: `ebw2143` November 22, 2019