Genetic Variations on the Game of Life

Norman H Packard and John S McCaskill

European Centre for Living Technology

Abstract

Conway’s Game of Life (GoL) cellular automata is extended to allow evolution through genetically encoded live states. In contrast with previous work, a deterministic selection mechanism is realized, so that mutation provides the only source of random variation in the dynamics. Systematic genetic variations near to the game of life rules are investigated and found to produce signs of computational complexity with an abundance of glider structures. Four levels of perturbations on the B3/S23 standard GoL rules are investigated: (0) genetic selection on the GoL, with the genes not influencing the GoL rules (1) genetic selection on GoL-like rules, e.g. S2gb3gB2g3 ∈ {S(2g(b))(3g(b))/B(2g)3(g)}, with conditional rules depending on genes denoted by g and survival rules involving birth overwrites denoted by b (2) genetic modulations of the live neighbour counting process that allow different numbers of live neighbours a) for specific 2nd neighbour ring configurations b) using masks on 1st neighbours encoded by the existing live neighbours c) with the gene specifying the allowed numbers directly (3) limitations of the influence of arbitrary gene encoded rule departures in varying symmetries through the requirement that state changes induced by a non-GoL rule inhibit further rule departures in their neighbourhood until corrected by a regular GoL rule. Whereas many perturbations of the GoL rules either quickly die out or proliferate rapidly to fill space with reproducing structures, a family of interesting dynamical systems is found, and its tendency towards open-ended evolution analysed by means of activity statistics. Note, that while the standard B3/S23 game of life, starting from random patterns of 50% 1s and 0s, in most instances does not produce complex dynamics, the new family does.

Notation: Rule extensions  [Golly](https://en.wikipedia.org/wiki/Golly_(program)) open-source cellular automaton package