cs6550

Brett Bonar

December 13, 2018

Abstract

Self-organization could be defined as the undirected movement of a chaotic structure into an ordered one. In a biological system, this may be seen as a formation of cells gradually reforms into a regular arrangement. Such arrangements could be lattices or grids that are identified as having regular or repeating patterns. Determining the behaviors with which to compose a self-organizing system is a nontrivial task. While many such systems have already been identified, the considerable amount of traits comprising cellular behaviors would make an exhaustive search of undiscovered systems impossible. The ability to discover these types of systems could prove valuable to disciplines such as medicine and bioengineering where regular patterns have real applications.

Introduction

The work done on this project is a continuation of a previous project for an Intelligent Systems course under Dr. Flann in 2015. The original goal of the project was to create a tool that is capable of identifying biological systems that organize into regular patterns. A biological system in this case is composed of one or more cell types and their behaviors. These cells are simulated using the CompuCell3D Simulator to determine if they are ”self-organizing”.

The idea is that these cells and their behaviors could then be replicated in the real world where their organized patterns may prove useful. Such cells could already exist in nature or be grown in a lab through genetic engineering processes.

The continuation of this project is to parallelize the most resource intensive operation of the application, which is the simulation of the cell growth.

Method

This project implements a simple hill climber written in Python that works on a limited set of cellular behaviors from the CompuCell3D simulator. An evaluation of the set complexity of the simulated result of a biological system is used to determine self-organization and regularity (i.e. the fitness function).

While CompuCell3D supports 3D cells, this simulation is limited to 2D for the purpose of simplicity and simulation speed. The CompuCell3D simulator supports a large number of cell attributes and behaviors. These have been narrowed down to a few variable parameters:

- Contact (adhesion) energy between cells
- Diffusion field parameters:

  Decay rate

  Diffusion constant

  Secretion rate for each cell type
- Chemotaxis for each cell type

The diffusion field and chemotaxis parameters are directly related. This project uses the CompuCell3D DiffusionSolverFE plugin to simulate chemotaxis. The secretion rate is the amount of chemical field that is secreted by each cell type. Chemotaxis then defines the rate at which cells are attracted to the chemical field. Chemotaxis is one of the most important and useful cellular properties for creating self-organizing biological systems.

The hill climber begins with a random configuration of the above parameters in addition to a random number of cell types (currently just one or two) and random volume for each type of cell. Number of cell types and cell volume are not changed between iterations. The initial configuration is run through the CompuCell3D simulator four times with different random seeds to generate four distinct results. Each simulation is run for 1000 Monte Carlo Steps (MCS). Each MCS step is one iteration of the simulation. The set complexity is then calculated using the four final outputs as the set.

The algorithm repeatedly performs a random walk on one of the variable parameters. If the resulting set complexity of the simulation is better than the previous best then this is taken as the new best result. This is repeated until no improvements have been made for a number of iterations, at which point the hill climber terminates and an XML file representing the latest best configuration is output.

The method used to calculate set complexity is as given by Dr. Flann and adapted to use image compression techniques on Dr. Flann's advice. The calculation of set complexity depends on the Kolmogorov complexity of a set of objects (in this case images output by the simulator), which can be estimated by the Normalized Compression Distance (NCD). The NCD can be considered a measure of the similarity of two objects. NCD is maximized when both inputs are random or dissimilar and is minimized when both objects are identical.

This implementation uses PNG compression on grayscale-converted images that are output by the CompuCell3D simulator. Images are concatenated for the NCD calculation by horizontally aligning them. There was a surprisingly significant difference in NCD results between vertical and horizontal concatenation. Horizontal concatenation was chosen because it gave a wider range of NCD values.

The benefit of parallelization of this algorithm is the ability to simultaneously simulate different random seeds for the full 1000 MCS. The previous algorithm simulated each of the four seeds sequentially to get the final result, which required significantly more time to complete. These simulations make up the bulk of the runtime of the application and are easily parallelizable as they may be completed independent of one another.

1 Parallelization

The hill climber was parallelized by splitting the simulation of each random seed of the