CISC499 Projects Proposed by Dorothea Blostein
Winter 2018

If you are interested in these projects email blostein@cs.queensu.ca. Please attach your transcript, or describe your background and courses you have taken.

Project 1: Create a library of tensegrity structures

The goal of this project is to create a library of tensegrity structures with a convenient interface for browsing or contributing new structures. When the user finds a structure of interest, he/she can download it and open it in PushMePullMe for physics simulation. For an example of how the library might be organized, see https://www.thingiverse.com/newest

Background information

A tensegrity structure is a network of isolated components under compression held in place by components under tension. For example, see Tom Flemons' models of leg, arm, torso, mast. More information at research.cs.queensu.ca/~blostein

PushMePullMe is physics simulation software written by mechanical engineer Gennaro Senatore. Several of my students have been extending the PushMePullMe user interface to improve the support for tensegrity construction. We would like to add modular tensegrity construction: a user selects tensegrity components from a library and selects a tensegrity combination method to create a larger structure. This 499 project is a first step toward that goal.

Project 2: Evolutionary computing to study models of evolution [Could be CISC499 or COGS499]

Develop software that allows the user to configure and execute various forms of evolutionary computing, with the goal of investigating various models of evolution. In particular, we are interested in comparing the diversity of species that results when evolution is modeled as "optimization" versus "satisfaction (finding a solution that is good enough)". This question arose last summer in discussions I had with an interdisciplinary research group that is investigating biopropulsion.

In evolutionary computing, a fitness function determines which organisms survive. When we are modeling evolution as optimization, the computation should continually select organisms that optimize the fitness function. In contrast when we are modeling evolution as satisfaction, the computation should allow an organism to survive as long as it is "good enough" (i.e. its fitness value is above a certain threshold). The fitness function must change over time as the evolutionary computation proceeds; this models that evolution takes place in a changing environment with changing survival pressures.

The time constraints of a one term project mean that you will need to restrict yourself to simulating the evolution of a *very* simple organism. Your 499 project should produce well-structured software that allows future students to substitute more complex organisms. I suggest that you represent a simulated organism as a floating point number; I am confident that interesting results can be obtained even with this very simple representation. Diversity in a population of these simulated organisms can be measured by the spread of values, where we consider nearby floating-point numbers as belonging to the same "species". The fitness function can be a weighted sum of several competing terms that select for things like "close to a prime number", "larger than K", "close to a multiple of 100". Change the weights in the fitness function over time, to model changing survival pressures. Various floating point operations can be used to create a new generation of organisms from the previous generation. My colleague in Mechanical Engineering, Professor Mike Rainbow, has a GPU cluster that could be used in this project.