Stanley Janesko

Genetic Algorithm Report

CS 461

Professor Hare

Genetic algorithms function my mimicking patterns in nature. Objects are scored on how well they will “survive” and the more highly rated members of the population will be chosen to breed first. Breeding involves the crossing over of one object data to another where there is a small chance of mutation. The genetic algorithm I created follows this natural pattern to create the most highly rated schedule.

The first object I created was a course object. This object contains a course name, and instructor, a time, and a room. Next, I created dictionary objects for the different values that a course may have. These included an instructor dictionary with which classes they could teach, a room dictionary with how big the classroom is, the expected enrollment for the course, and a time slot array. These allow for easy look up while creating and checking the fitness functionality of a schedule. The core structure of this program is the schedule. The schedule uniquely contains one of each course (a total of 12). This schedule object is created populationSize times and added to an array. This population object is where the fitness function, the normalization and cumulative distribution occur.

I did not get the desired results I imagined because I believe the fitness calculation could use a few changes. I believe an instructor teaching a class that they can teach should be more weighted than, let’s say, the room size being able to accommodate the expected course enrollment. After toying with my fitness function for a few hours and making changes where I saw fit, I was not able to produce results that increased the chance of a better schedule. Schedule creation is more complicated than it lets on and will require further work to create the best fitness calculation.