TRINITY COLLEGE DUBLIN, SCHOOL OF MATHEMATICS

Assignment #2, Module: MA5611

Gustavo Ramirez

January 20, 2017

1 Problem description

The goal of this assignment is to develop a framework for implementing genetic algorithms in parallel using MPI. Then using this framework you will develop strategies for playing the iterated prisoner's dilema.

1. Task 1

2. Task 2

3. Task 3

1.1 Solution

- **Task 1**

  For the implementation of the genetic algorithm:

  1. generate the population with its values randomly set. This is only done once
  2. calculate the fitness for all the members of the population combined and separated
  3. calculate the probabilities of being selected, each of them proportional to their fitness, and normalized. These probabilities are useful in the next step, in the sense that when a chromosome is to be selected, it is done through the wheel roulette selection.
  4. crossover step. X% of crossover rate means that percentage of the next generation is composed of crossover offspring (and for creating two new elements of the offspring, two elements of the current generation are chosen through wheel roulette selection and then crossed), and the rest of the empty places are filled in doing wheel roulette selection again. The two chromosomes from the current generation crossover after first bit to give the new offspring. **If the population is n, and the fractional crossover rate is c_r, then this step loops nc_r times; in each iteration of this loop, a pair of current chromosomes is selected, crossed, and the offspring created and stored. Additionally, it loops $n - nc_r$ more times, to fill the empty places.**
  5. mutation step (acts over the just created offspring). Y% of mutation rate means to flip each bit with a probability Y%. **If the population is n, the chromosome size s, and the mutation rate m_r, then this step loops nsm_r times, but without flipping two times the same bit. For this flipping process, all of the ns bits of the population are taken into consideration.**

  Perform steps 3, 4 and 5 n_g times, where n_g is the number of generations.

- **Task 2**

- **Task 3**