CS 472 EVOLUTIONARY COMPUTATION

Assignment 1b

Alex Cochrane

February 19, 2014

1 ABSTRACT

The problem for this assignment was to create a Genetic Algorithm for the spherical function used in the first assignment. To create a population for this problem, I decided to model my population with a class. The class then held on to an array of 1000 Individuals, my population, who had their fitness and vector stored with them. The Class Diagram for my population model is as follows.

2 ALGORITHM DESCRIPTION

For this assignment I focused solely on Tournament Selection even though I added a method for Roulette Selection under my Population class. The pseudo code for Tournament Selection is as follows:

```
while ( Individual[]->fitness > 0.01 ) //larger then the smallest value measured
    if( repeated fitness 100 times)
        break
    while( nect Population is not filed )
        create a new individual to put in the population
        randomly select an individual from the current population N = 5 times
        select the one with the best fitness and make the new individual it
        mutate by + or minus 0.01 on the ENTIRE vector
        calculate the new fitness / sort
```

Something I did that could possibly be of use later is sorting my population so that the first element in my individual array is always the best fitness. I then used some mathematical maneuvering to get my random selection function to pick from lower numbers more often. This gave my selection some sort of a ranking and seeding aspect which I believe got better solutions with every selection.

When the process is over it prints the best solutions fitness which should be below 0.01 unless it got stuck with a bad population that could not get to zero. It also prints how many times it iterated just to get a feel of how changes made the process more efficient or not.

---

3 Results

The average amount of time spent generating populations was around 450 iterations per run of Tournament Selection. Fitnesses of the best individual in each population is always below the specified amount, meaning it is always finding the minimum. With the sphere being such a simple case however, this is to be expected. Nonetheless it shows how well the Genetic Algorithm can work at finding the best solution because of its use of the best individuals each round. This is especially true with the sorting and seeding that I do which seems to decrease the number of generations by about 20 to 50 for this problem. The following two pictures show some values coming from a sample run and the graph of the values.

---

4 Conclusion

With the behaviour of the Genetic Algorithm working well at selecting and mutating the populations, it is hard to find something wrong with the way everything is working. This is also in part because it is being used to solve the trivial sphere function. I would say the only way to improve the GA at this time would to alter the mutation method or add crossover. Testing on the sorting of the populations could also be done.
