BioComputation

Worksheet Optimisation

For the optimisation thread of the assignment the idea is for you to carry on from worksheet 3 by implementing some other operators and to show competitive performance on a variety of numerical functions.

For example, in the lectures we have talked about roulette-wheel selection where individuals have a chance to be selected based upon their fitness *relative* to the others in the population. You could implement roulette-wheel selection in your GA and compare its performance to tournament selection on various functions. For maximisation functions:

- Pick a random number between 0 and the total fitness of the current population.
- Starting at the first individual, maintain a running total of the population fitness until the randomly chosen fitness value is met or exceeded.
- Add a copy of the individual whose fitness was last added to the offspring population.
- Repeat P times.

```
for( i=0; i<P; i++ ) {
        selection_point = random()%population_fitness_total;
        running_total  = 0;
        j=0;
        while ( running_total <= selection_point ) {
                running_total += population[j].fitness;
                j++;
        }
        offspring[i] = population[j-1];
}
```

Both the mutation and crossover processes can be altered for use with reals, as hinted at in the first artificial neural network lecture. You could research, implement and explore some of them.

The two functions on worksheet 3 are not particularly challenging, the real-valued version of the counting ones task is very simple in particular. Here are two more minimisation functions:

$f(\boldsymbol{x}) = \sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(1-x_i\right)^2\right]$ where -100 ≤ *x* ≤ 100, start with *n*=20	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}cos2\pi x_i\right)$ where -32 ≤ x ≤ 32, start with *D*=20