Bio-Inspired Artificial Intelligence - EMATM0029
Corrections - Artificial Evolution

In case of questions, don't hesitate to contact sabine.hauert@bristol.ac.uk.

Exploring Artificial Evolution

Evolving a Cellular Automaton

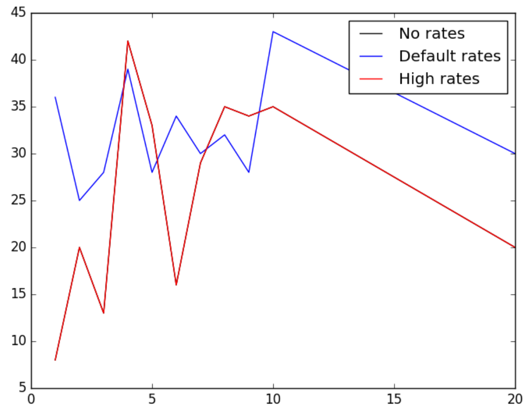

Figure 1: Best fitness value (y axis) for each generation (x axis) with mutation and crossover rates set to 0, to the default rates, and to high rates.

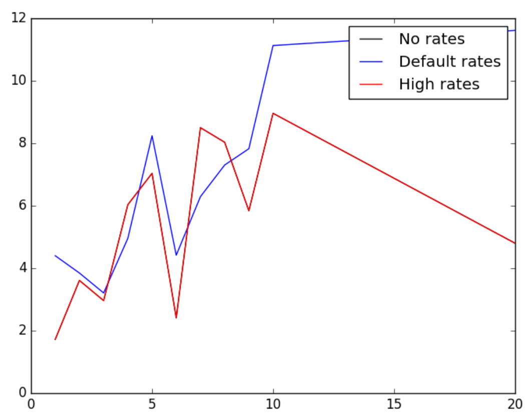

Figure 2: Average fitness values (y axis) for each generation (x axis) with mutation and crossover rates set to 0, to the default rates, and to high rates.

1) High scores: With no mutation/crossover, the selection process is continuously selecting from the surviving individuals of the last population, and with little variation introduced the scores tend to fluctuate. With mutation/crossover, the high score tends to be steadily increasing, albeit at a

slow rate.

2) Why is the average score increasing with a zero mutation/crossover rate? The selection of individuals for a new population is based on the fitness of individuals in the previous population, and those that perform well (receiving a greater fitness value) are more likely to be copied to the new population (higher probability for selection); often multiple times compared to a potential for individuals of less-than-average fitness to not be copied at all. This increases the average fitness overall.

3a) High mutation/crossover: There is a huge difference between the previous experiments and the new experiment with high mutation/crossover rates. The higher rates produce lower high scores and lower average scores. The high mutation and crossover rate means that you typically lose the "genes" that produce the high fitness scores for individuals, so even if individuals with high fitness levels exist in one population, and are therefore very likely to be copied to the next population, these copies will likely experience mutations, and will all be crossed with other individuals in the population. This may lead to a loss of the individual which had previously received a very high fitness rate.

3b) You would expect the diversity to be much higher, as mutation is taking place fairly often and crossover always occurs, introducing new individuals from the mixing of previous individuals, and producing entirely unseen combinations through mutation.

4) No, high-scoring individuals are not always preserved from year to year in the experiments where mutation and crossover exist. Elitism could be used instead to preserve the best individuals from generation to generation.

Evolving a Walker

1) No, the genomes do not change over time, as is evident by the names which are based on the genomes. This is because all 20 are being copied over in the current setting.

2) Yes, with the introduction of new individuals we see the performance quickly increasing over time.

3) The population shows signs of convergence, with similar letters appearing in the same place (V as the first letter of the first name in my example, and mostly Z or Q as the first letter of the last name). However it has not yet converged to very similar names altogether, as it does around the 50 generation mark.

4) Diversity can be increased by increasing either (or both of) the mutation probability and mutation amount, causing the individuals in the population to be mostly unique. Diversity can also be greatly decreased by reducing these to 0.

Designing an Evolutionary Algorithm

1) Solving a maze: the genome could be an array of turn decisions which has a maximum length of 10, e.g. [R L L L R R L]; this maps directly to the path taken by the robot (the phenotype). In order to promote fast solutions the fitness function could consider distance remaining to the exit over time. One example fitness could be $fitness = \sum_t^{duration} \frac{1}{duration} maxDistance - distanceRemaining$

2) Summing 3 numbers: the genome could be a tree which represents the operations to be applied to each input variable and hence a mathematical expression (the phenotype). The fitness function should consider how close the expression result is to the desired result, but could also consider the number of operators used (we know the ideal result is simply 3 uses of '+'). One example fitness could be $fitness = \frac{largeNumber - (valueSum - expressionResult)^2}{largeNumber} + \frac{1}{numDifferentOperators}$

3) Designing a cellular automaton: the genome could be a binary sequence representing the rules of your automaton. For example, if you are considering a von Neumann neighbourhood then you have four neighbours to consider. Your genome should list whether your cell lives or dies (1 or 0) for each combination of neighbour states possible (0000, 0001,0010...). The fitness function should minimise the distance between agents. Assuming agents can be on the same square, you could for example minimize the amount of black squares (1 value). For example $fitness = \frac{1}{NumBlackSquares}$

Impact of Parameters

1) Making your problem more difficult for most agents will lead to the few remaining agents achieving high fitness, therefore leading to high selection pressure.

2) Increasing the mutation probability will increase diversity as 'new' (different) individuals will enter the population at a faster rate. The impact on finding a good solution depends at what stage we are at in the evolutionary process; near the beginning increased diversity is generally good as we get access to more solutions, one of which might just by chance be much better than where we are at now (we haven't explored the fitness landscape much yet). However, as we use evolution to navigate that landscape and hone in on the best solutions, random mutations might throw us 'backwards' and undo that work.

3) The difference in diversity resulting from proportional and rank based selection is most apparent in situations where there are a few high fitness individuals. In this case proportional selection will lead to those individuals getting selected very often; in rank based the lower fitness individuals have a greater chance of being selected as it's based on overall rank rather than specific fitness. How this impacts on the ability to find a good solution again depends on where we are in the evolution of our solution; we do want to converge towards a good solution eventually however the extra diversity of sometimes picking a lower fitness individual may take us to even better solutions.

4) As tournament size increases selection pressure also increases. This is because we only pick the top X individuals and so the more individuals there are in the tournament, the less likely it is to be one of those top X.

5) Increasing population size does improve your ability to explore rugged fitness landscapes IF

(a big if!) you use all of those extra population members and maintain diversity for a decent while before homing in on a solution (think about selection methods, mutation etc). It's no good having a large population which very quickly converges to a solution!

Debugging Evolutionary Algorithms

The lack of improvement in fitness suggests a few possibilities: - we simply haven't seen a good individual yet
- we aren't selecting good individuals for reproduction
- our reproduction methods aren't generating 'children' that are fitter than their 'parents'.

Addressing each of these in turn:
- we could increase the population size or increase the diversity in the population through an increased mutation rate, simply making it more likely that we have some good individuals in the population
- we could change our selection method; specifically we could try to hone in faster on good solutions so perhaps use a stricter method such as proportional selection
- we could change our reproduction method to change how we combine selected individuals; it's always good here to have a sanity check and make sure that the way you are recombining genomes actually makes sense