NAT Tutorial 3 (week 5)

1. [Baldwin effect] It has been observed that some organisms seem to pass on behaviours learned during their lifetime to their offspring. Lamarck's hypothesis was that traits acquired during the lifetime of an individual could somehow be passed on genetically to the individual's children. Until the recent past, however, Lamarck's hypothesis was universally rejected, because there seemed to be no obvious biological mechanism for this. Today, one admits that Lamarckian inheritance is in principle not impossible and that Lamarck's theory of evolution and development cannot be reduced to the inheritance of acquired traits.
   One proposal for a non-Lamarckian mechanism explaining the passing on of learned behaviours was given by Baldwin, who pointed out that if learning helps survival, then the organisms best able to learn will have the most offspring (highest fitness). Further, if the environment remains constant, so that the best things to learn remain constant, then this can lead, via selection, to a genetic encoding of a trait that previously had to be learned. Describe how you could use evolutionary computation as a model system to demonstrate the truth (or otherwise) of Baldwin's hypothesis.
   Another important feature of Baldwinian evolution is the identifiability of isolated fitness maxima. How would this be possible?

Answer: In the lecture the paper by Nowlan and Hinton was mentioned with gives an example (with random search instead of a GA, but this does not make a difference in the all-ones problem as we have seen before). In addition, any GA that includes local search can be interpreted as an example for Baldwinian evolution if the result of the local search is used to edit the individual's genetic code. The second question is related to "Gaussian evolution", i.e. the increase of the area searched by a species by random (Gausian) variations.

2. (Mitchell) Design a three bit fully deceptive fitness function. "Fully deceptive" means that the average fitness of every schema indicates that the complement of the global optimum is actually the global optimum. For example, if 111 is the global optimum, any schema containing 000 should have the highest fitness in its partition.

Answer: One possibility is $(1-x)+(1-y)+(1-z)+4*xyz$, where x, y, z are the three bits. Except if there are only ones switching a one to a zero increases the fitness. Consider the ability to fly as an example: In order to fly the individual needs simultaneously: wings, a good brain for control, specific muscles, some stiffening of the body etc. All these features have a cost (negative fitness points), only if all of them are there and the phenotype can actually lift off then a fitness increase is reached. Evolution will thus have to use properties that have already proved useful (but perhaps in a different context) instead of waiting for a very lucky mutation of several bits at once.

3. How can elitism be achieved in an application of GA to multi-objective optimisation? Visualise the situation for the case of two fitness functions. What issues can be expected to arise for a large numbers of objectives?

Answer: The simple answer is to spare the first front. If the algorithm is already in the converging phase this might slow down the progress (no "killer instinct") such that one may rather keep only one a few individuals from the first front. The algorithm discussed in the lecture, assigns a distance function to the individuals which may be used for this subsection from the first front, i.e. only those with the

largest distances will survive. On the other hand, does this particular algorithm keep the full parent generation together with a (mutated) children population and performs selection on this double population such that elitism is already included.
For a large number of objectives there may be very many (most) individuals on the first Pareto front (which is of dimension m-1 for m fitness functions) such that there is not much competition. Either one has to work with very large populations or one has to think about selecting a few fitness functions for the non-dominating search while the other ones are combined by some idea about there priority. Discuss parallels to natural evolution.

4. A (1+1)-ES performs essentially only hill climbing. Give an estimate of the time to reach the minimum of the function $f(x)=(x-5)^2$ when starting from $x=0$. This estimate will depend on the size of the mutation steps and the required accuracy.
   What strategy would you choose for Rastrigin's function in n dimensions $f(x)=10\,n + \Sigma\,(x_i^2 - 10\cos(2\,\pi\,x_i))$ [the sum runs from i=1 to n]. Discuss the dependency of μ and λ on n (start with considering n=1) for a (μ,λ)-ES and a (μ+λ)-ES and reasonable choices of the parameters and rules for mutability.

Answer: If we keep the mutation size constant and the required accuracy is d then we should set the mutation rate to d (standard deviation for Gaussian mutations) such that will need about 10/d step where the factor 2 is due to the fact that in about half the cases the mutation is towards the correct side (otherwise we keep the parent). Note that it might be a good idea to use a modifiable mutation rate. How should the mutation rate be changed or how would it change when being evolved together with x? For Rastrigin's function a strategy with evolving mutabilities would probably lead to mutations of size one which allows the individuals to jump from one minimum to the next one. However, larger mutation sizes may evolve as well such that the global optimum could be missed. This can be prevented by the 1/5 rule, which leads to an increase of the mutability when it is possible to reach lower minimums, but to a decrease when only smaller steps lead to a further improvement. Make sure that a maximal mutation size is set which can be usually be derived from the problem specification. μ (number of parents) should be large enough to ensure that a non-optimal individual in a better minimum can still survive, while λ needs to be large enough to guarantee that out of the increasing number of neighbours the good ones can be found, i.e. both need to grow linearly with n.

5. Genetic programming (GP) is an evolutionary technique which attempts to evolve programs fit for some purpose. Describe a typical GP system: explain how programs are represented in the system; give examples of the genetic operators applied; and state the main steps of the evolutionary algorithm indicating where there are design choices to make.

Answer: s. lecture slides

6. Express the following functions in Lisp notation, using only + − / as non-terminals and *x*, 0, 1, 2, 3, . . . as terminals.
   a) $y = 3x + 2$
   b) $y = 5x^4 - 2x^2$
   c) $y = -0.25x^3 + 3.5$

   Which of these functions can you represent using only *x* and 1 as terminals?

Answer:

a) (+ (* 3 x) 2)

b) (- (* 5 (* x (* x (* x x)))) (* 2 (* x x)))

c) First, rearrange: y = 7/2 - [x^3]/4

(- (/ 7 2) (/ (* x (* x x)) 4))

7. What fitness function can GP use for solving symbolic regression problems? Can you think of any alternatives? How much domain specific knowledge about the problem is encoded in this fitness function?

Answer: A simple one could be: 3 points for getting an answer correct, 2 points if you are within 1% of the value, 1 point for getting within 5% of the value. Alternatives: least squares, etc. Knowledge: quite a lot, since this can't be applied generally and we are telling the GP system to find nearest-fit functions and combine them.

8. Do schemata and building blocks exist in Genetic Programming populations?

Answer: An open question! Clearly the program (+ (* x x) #) where # is the don't care symbol is a sample of all programs of the form x^2 + #. The question is whether it makes much sense to talk about the average fitness of such a pattern. Some restricted forms of GP, such as the evolution of a set of production rules may well have building blocks in the population: a BB is a high-fitness rule (which may not work in all contexts but will usually lead to *some* improvement of the fitness). For more arguments see Banzhaf et al's Intro to GP book, or Ricardo Poli's work on a schema theorem for GP.