Evolutionary Computing
(Genetic Algorithms)

- Heralded as an approach to machine learning
  - Learning new solutions/behaviours by evolving old ones
- Also an approach to search
  - Search in a potentially infinite search space
  - No guarantees of success
- Inspired by theory of evolution
  - Based on fundamental genetic processes
  - Not constrained by them though
- Examples
  - 1. Selecting a present for somebody
  - 2. Shakespeare and the monkeys
  - 3. Fitting equations to data points

Example 1 : Selecting a present for somebody

- You enter a large shop with no idea what present to buy
- You'll know it's right when you see it
- Ideas develop as you browse
- Sometimes you **leap** from one idea to a totally different one
- Sometimes you **combine** one or more previous ideas
- Gradually you **refine** your choice
- *[Eventually you run out of time and buy something terrible!]*

Example 2 : Shakespeare and the monkeys

- Dawkins (1986) considered a single line from *Hamlet:*

  METHINKS IT IS LIKE A WEASEL
- The probability of generating this line of 28 characters (including the spaces) from the 27 character alphabet by chance is (1/27)^28
- Or, put another way, after 27^28 attempts you could expect to produce it just once (after millions of years)
- Dawkins wrote a GA program which generated the line in between 41 and 64 attempts (taking about 11 seconds)
- Instead of **single-step** selection the program used **cumulative** selection

Example 3 : Fitting equations to data points

- Consider the equation for a straight line between 2 points:

  $y = mx + c$
- Given the points (*x1*,*y1*) and (*x2*,*y2*) we can determine *m* and *c*
- Using a GA we could start off with random values for *m* and *c* and gradually **evolve** better and better values for *m* and *c* by making small changes and **breeding** from the best until the error is acceptably small
- This is a trivial problem but with larger data sets (more than just 2 points) the GA approach offers a potential route to a solution

Three Main Forms

Three main forms of EC are distinguished–

- Genetic Algorithm (GA)
  - The classical form
  - Evolving new states from old during search
- Genetic Programming (GP)
  - Evolving computer programs
  - Using GAs to evolve source code or representations thereof
- Evolutionary Strategies (ES)
  - Probabilistic
  - Mutation-based search
  - cf. stochastic random walk

Key Elements of a GA

- Natural Selection
  - A fitness measure determines which population members (solutions) survive
- Reproduction
  - Creating a successor generation by ...
- Crossover of chromosomes
  - Combining two or more “parents” to form “offspring”
- Mutation of genes
  - Introducing aberrant offspring at random intervals
- Probability
  - All choices are probabilistic

The GA Method

- A population pool is created
  - This contains possible solutions
- A fitness function is applied to each individual
  - This determines how “good” each solution is
- Individuals are selected for a mating pool probabilistically
  - The fitter the individual the more likely it is to be selected
- Individuals in the mating pool are combined using crossover
  - Again selection is probabilistic
  - Crossover points are also selected probabilistically (mutation may occur)
- The fittest individuals become the next generation

Fitness Functions

- A quality function rates each individual’s “goodness”
- Fitness is commonly normalised to lie between 0 and 1
  - The fitness of an individual is then the quality of that individual divided by the total quality of all individuals in the population
- Identifying fitness functions is one of the more difficult tasks in evolutionary computing

Selection Schemes

- Rank
  - The fittest individuals in the current generation are chosen for the mating pool
- Roulette
  - Fitter individuals are allocated a larger area of the wheel
  - Thus they are more likely to be chosen than less fit individuals
- Tournament
  - Pairs of individuals compete for entry to the mating pool by comparing their fitness scores
  - Successive rounds of pairing off can further reduce the number of candidates for the mating pool
- Elitist
  - The fittest individual(s) always proceed to the next generation
  - By-passing the mating pool

Reproduction

- Reproduction creates candidates for the next generation of individuals by recombining elements from the current generation which have been selected for the mating pool
- Genetic recombination produces offspring
  - Part of the genome of each parent is passed on to the offspring
  - In humans one half of each chromosome pair is passed on by each parent
- Mutation of genes can occur during recombination
  - This is very rare
  - You can envisage mutation as changes within the substrings

Crossover

Individual A	a1	a2	a3	a4	a5	a6
Individual B	b1	b2	b3	b4	b5	b6

Pick a crossover point, say 4

Offspring 1	a1	a2	a3	a4	b5	b6
Offspring 2	b1	b2	b3	b4	a5	a6

Mutation

a1	a2	a3	a4	b5	b6

Select gene to mutate, say 3

and value to mutate it to, say c3

a1	a2	c3	a4	b5	b6

EC Bibliography

- Dawkins, R., 1986, *The Blind Watchmaker*, Penguin Books.
- Fogel, L., Evans, M. & Walsh , M., 1966, *Artificial Intelligence through Simulated Evolution*, John Wiley.
- Goldberg, D.E. & Holland, J.H., (editors), 1988, special issue of *Machine Learning*, **3** (2/3).
- Goldberg, D.E., 1989, *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley.
- Holland, J.H., 1992, *Adaptation in Natural and Artificial Systems*, 2nd edition, MIT Press.
- Koza, J.R., 1992, *Genetic Programming*, MIT Press.
- Rechenberg, I., 1971, *Evolutionsstrategie - Optimierung nach Prinzipien der biologischen Evolution*, Dr.-Ing. Thesis, Technical University of Berlin.