Evolution Strategies
Evolutionary Programming
Genetic Programming

Michael J. Watts

http://mike.watts.net.nz

Lecture Outline

- Evolutionary Algorithms revision
- Genetic Algorithms revision
- Evolution Strategies
- Evolutionary Programming
- Genetic Programming

Evolutionary Algorithms

- General purpose algorithms
- Based on some aspects of biological evolution
- Useful for solving multi-parameter optimisation problems
- NOT random search algorithms

Genetic Algorithms

- AKA GA
- Based on populations of artificial chromosomes
- Solution attempts are encoded as value strings in the chromosomes

Genetic Algorithms

- New individuals created by two mechanisms
  - crossover (recombination)
  - mutation
- Selection based on fitness of chromosomes
  - stochastic selection

Evolution Strategies

- Invented early 1960s in Germany
- Ingo Rechenberg, Hans-Paul Schwefel and Peter Bienert
  - engineering students
- Experimenting with wind tunnels
  - optimising jointed flat plates
- Only intuitive methods to do this at the time

Evolution Strategies

- Rechenberg had the idea of 'mutating' the parameters and selecting good mutations
- ES are used for numerical parameter optimisation
- Parameters of a problem are encoded as real numbers
  - real-numbered chromosome

Evolution Strategies

- Fitness of individual is determined by how well the parameters solve the problem
- Offspring are created by mutation
- Real-numbered, normally distributed creep mutation
- Offspring replace parents only if more fit

Evolution Strategies

- ES named according to number of parents and children at each generation
- 1+1 ES has one parent and one child

1+1 ES

1. Evaluated fitness of parent *P, f(P)*
2. Create child *C* by adding small normally distributed values to each parameter of *P*
3. Evaluate the fitness of *C, f(C)*
4. If *f(C) > f(P)* then
   replace *P* with *C*
5. Repeat Steps 2-4 until stopping condition

Evolution Strategies

- Later ES have populations
- (m+l) and (m,l) ES
- m (mu) is the size of the parent population
- l (lambda) is the size of the offspring population
- offspring are created using recombination as well as mutation

Evolution Strategies

- In a (m+l) ES, the m best survive to the next generation
- In a (m,l) ES, only child individuals survive to the next generation

Evolutionary Programming

- Invented early 1960s in the USA
- Created by Lawrence Fogel
- Regarded artificial intelligence as the ability to predict a symbol based on previous symbols
- Evolved a population of finite state automata to perform this prediction

Evolutionary Programming

- Evolution consisted of adding, modifying or deleting state transitions
- Task was to predict characters from streams of characters
- FSA with the least number of errors were allowed to reproduce

Evolutionary Programming

1. Create a population of solutions
2. Evaluate each solution in the population
3. Select individuals to reproduce
   - tournament selection
4. Mutate the reproduction population
5. Repeat 2 - 4 until stopping condition is reached

Evolutionary Programming

- Reproduction in EP is via mutation
- Mutation may be normally distributed
- No prescribed method of representation
  - Use whatever works for the problem
    - FSA, ANN etc
- Crossover / recombination is not used

EP and GA

- Main differences between EP and GA
- GA requires the solution attempt to be encoded in a string of values
  - genome
- EP uses whatever representation fits the problem

EP and GA

- Mutation in EP is a normally distributed perturbation
- Has infrequent large changes, frequent small changes
- Mutation rate decays as run time elapses
- GA mutation tends to be fixed size changes that create entirely new values

EP and ES

- EP and ES are very similar
- Major differences are selection and recombination
- Selection in EP is stochastic
  - tournament based
  - randomly selected participants

EP and ES

- Selection in ES is deterministic
  - bad individuals are purged
  - good individuals breed
  - no randomness involved
- No recombination is used in EP
- Multi individual ES will use recombination

Genetic Programming

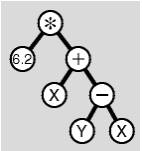
- Created by John Koza
- Use an EA to create algorithms
- Score attempts based on
  - how well the problem is solved
  - how efficiently it solves it

Genetic Programming

- Mechanistically similar to GA
- One major difference
  - no genotype / phenotype distinction
  - evolutionary operations carried out directly on the candidate programs themselves
- Reproduction is via crossover and mutation

Genetic Programming

- Candidates are represented as parse trees
- e.g.



Adapted from http://www.genetic-programming.com

Genetic Programming

- Crossover is implemented by swapping branches of the tree
- Mutation involves deleting and re-growing branches

Summary

- ES, EP and GP are all different kinds of evolutionary computation
- Each developed separately, but have common themes
- The boundaries between each are not clear-cut!
- Each have their own niches

References

- “Evolutionary Computation: Towards a New Philosophy of Machine Intelligence” by David B. Fogel
- “Genetic Programming” by John Koza