VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Engineering and Computer Science

COMP 307 — Lecture 10

Evolutionary Computing 1 (ML7)

Evolutionary Computing and Learning

Dr Bing Xue (Prof. Mengjie Zhang)

bing.xue@ecs.vuw.ac.nz

Outline

- Why evolutionary computing and learning?
- Where does EC come from?
- What is EC about — EC Techniques
- Main Idea
- Evaluating candidates
- Genetic algorithms: representation, selection and genetic operators
- Overview of other evolutionary algorithms

Why Evolutionary Computing and Learning

- We have discussed a number of methods and algorithms in Machine learning
- What are they?
- Major characteristics/limitations
  - A single solution over an experiment run
  - Local optima
  - Unreasonable assumptions
  - Define the structure/model of the solutions, then learn parameters/coefficients
  - Large structure of the learning machines/solutions for high dimensions of input features
- Are there any ways for avoiding the problems/improving situations?

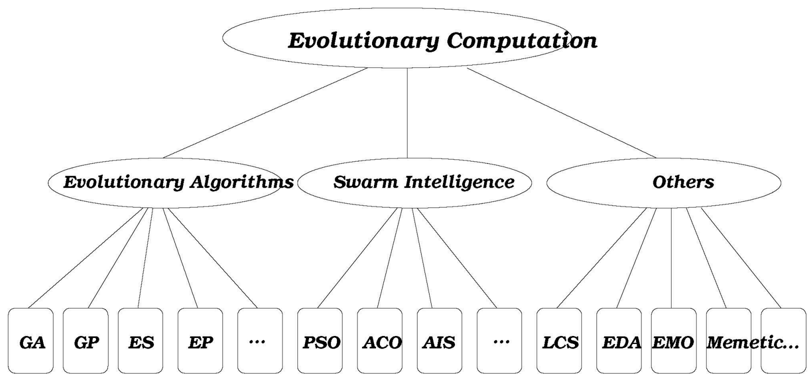
Evolutionary Computing — Origins

- Evolutionary computing techniques
- In the 1950s, long before computers were used on a great scale, the idea to use Darwinian principles for automatic problem solving originated.
- Three different interpretations of this idea were developed independently
  - Evolutionary programming: Lawrence Fogel (USA)
  - Evolutionary strategies: Ingo Rechenberg (Germany)
  - Genetic algorithms: John Holland (USA)
- These areas developed separately for over 15 or 20 years.
- Since the early 1990s, they have been seen different representatives of one technology, *evolutionary computing/computation*

EC Techniques

- EC techniques mostly involves meta-heuristic optimisation
- Evolutionary algorithms
  - *Genetic algorithms* (the biggest brunch)
  - Evolutionary programming
  - Evolutionary strategy
  - *Genetic Programming* (Koza, 1990s, fast developing area)
  - Learning classifier systems
- Swarm intelligence
  - Ant colony optimisation
  - Particle swarm optimisation (PSO, a fast developing area)
- Other techniques
  - Differential evolution
  - Artificial life
  - Artificial immune systems

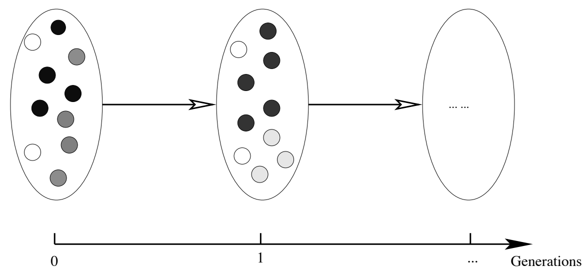
EC Techniques



EC Key Idea

- Biologically inspired
  - Like NNs, but source is evolution, not neuroanatomy
  - Process: Reproduction/elitism, recombination, mutation
  - Natural selection
  - Survival of the fittest
- Search for an optimal solution in the way evolution searches for optimal species
- Parallel search with a ***population*** of individuals
- Stochastic
  - Changed pieces of information randomly chosen
  - Individuals with a better fitness have a higher chance to be selected, but typically even the weak individuals have a chance to be chosen

Search in Evolution

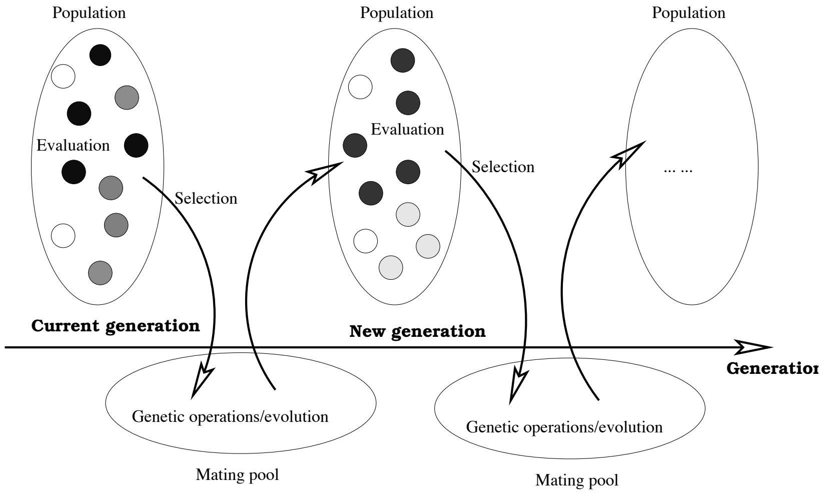


- Generation by generation
  - Some reproduce
  - Some die
  - Some newly produced

Evolutionary Search

- Search space of *candidate solutions*
  - Not space of *partial solutions*
  - *Modify* whole solutions rather than *extending* partial solutions
- *Genetic beam search*
  - Keep track of a set of good solutions
  - Not all candidate solutions, like best first or A*
  - Not only the best candidates, like in hill climbing or gradient descent
- Need way of evaluating the quality of solutions
- Combine candidates to construct new candidates
  - Not just modifying candidates in isolation
  - Different candidates can interact in evolution

Evolutionary Search



Evolutionary Search

- The current generation
  - A population of candidate solutions
  - Evaluation
- Evaluation: Fitness function/evaluation
  - Performance measure of candidate solutions
  - Competition
- Selection: Population −→ Mating pool for evolution
  - Select good candidates
  - Selection pressure
- Evolution: Genetic operators
  - Retain/copy (elitism/reproduction): not getting worse
  - Recombination (crossover): Improve candidates
  - Mutation: maintain diversity of population
- New generation(s)

Evaluating Candidates

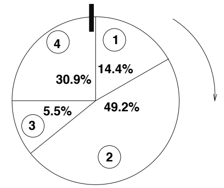
- Need measures of quality of candidates
  - Must correspond to desired optimality property
  - May have to apply to bad solutions (as well as good solutions)
  - Must be computable
  - Need to be smooth:
    - large changes to candidates −→ large changes to quality/fitness;
    - small changes to candidates −→ small changes to quality
- The term *fitness* is usually used to represent the quality of a candidate
- The measure is usually called *fitness function*
- Depending on the task, the fitness function can be designed:
  - the larger, the better --- maximisation
  - the smaller, the better --- minimisation

Representations: Genetic Algorithms

- A large brunch of evolutionary computation
- Since 1970s, there are a number of different representations
- The standard representation
  - Candidate solutions (individuals in the population): bit strings, encoding solutions to bit strings
  - *Chromosomes*
  - Crossover and mutation operate on substrings of the bit string
  - Random vs nonrandom (e.g. uniform)
- Candidate solutions (individuals in the population): *floating point numbers*
- Also some other representations recently

Genetic Algorithms: Roulette Wheel Selection

- For each member of the population, allocate space on the roulette wheel in proportion to fitness.



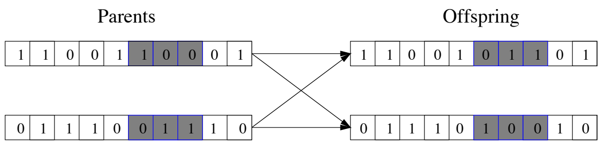
- Spin the wheel and put string where it stops into the *mating pool* — a tentative/temporary population
- Repeat until the mating pool is full
- This strategy ensures that the fittest individuals are more likely to be selected for reproduction

Genetic Algorithms: Genetic Operators

- Crossover
  - One point crossover

    Parent1: 0111|1 Child1: 01110

    Parent2: 1100|0 Child2: 11001
  - Two point crossover: for long chromosomes



- Mutation
  - Randomly modify a particular bit of the selected individuals
  - Main goal: maintain the diversity in the population

Tackling A Problem with GAs

- Formulate the problem as find min/max of $f(p_1, p_2, p_3, \ldots, p_n)$
- Ensure that $\boldsymbol{f}$ can be evaluated for all values of p_i
- *Encode* the $\boldsymbol{p_i}$ as binary strings
- Define/use selection and genetic operators
- Determine the GA parameters
  - Population size
  - Crossover rate
  - Mutation rate
  - Stopping criteria
- Feed into a GA 'engine' and wait until it stops
- *Decode* the solution

GA Applications

- Numerical Optimization
  - Design of jet engine turbines. 100 parameters. Which values are best? Boeing 737
  - Finding weights of a neural network
- Combinatorial Optimization
  - Glass cutting
  - Time tabling and job shop scheduling
  - Bin packing, Beer pallet loading
  - Scheduling of aircraft arrivals
  - National basketball league draw
  - Distribution, Travelling sales person (TSP)
- Data Mining, classifier learning
- Face detection, image and vision applications
- Genetic art, movies, Robocup

Evolutionary Computing Techniques

- Particle swarm optimization (PSO):
  - http://en.wikipedia.org/wiki/Particle swarm optimization
- Learning Classifier Systems:
  - http://en.wikipedia.org/wiki/Learning classifier system
- Ant colony optimization:
  - http://en.wikipedia.org/wiki/Ant colony optimization
- Differential evolution:
  - http://en.wikipedia.org/wiki/Differential evolution
- Other useful links:
  - http://en.wikipedia.org/wiki/Genetic Algorithm
  - http://en.wikipedia.org/wiki/Evolution strategies
  - http://en.wikipedia.org/wiki/Evolutionary programming

Summary

- Evolutionary computing overview
- Main idea and process
- Representations of candidate solutions
- Selection and genetic operators
- Genetic algorithms
- Other EC algorithms and techniques

- Next lecture: Genetic programming