



School of Engineering and Computer Science

COMP 307 — Lecture 10

Evolutionary Computing 1 (ML7)

Evolutionary Computing and Learning

Mengjie Zhang (Meng)

mengjie.zhang@ecs.vuw.ac.nz

COMP307

EC1 (ML7): Overview : 3

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?

COMP307

EC1 (ML7): Overview : 2

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

COMP307

EC1 (ML7): Overview : 4

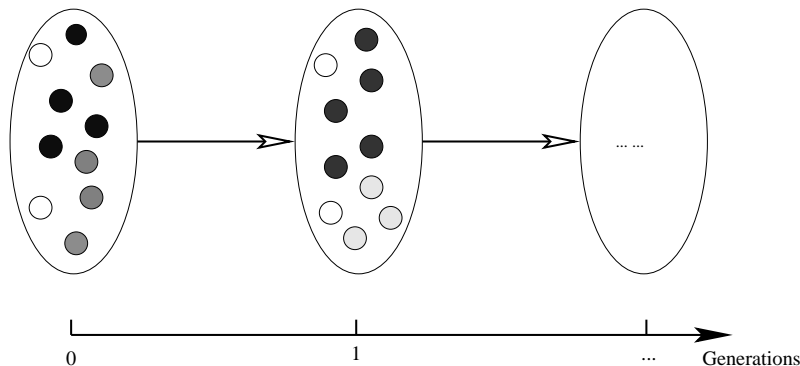
Evolutionary Computing — Origins

- 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
 - ...

Search in Evolution



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

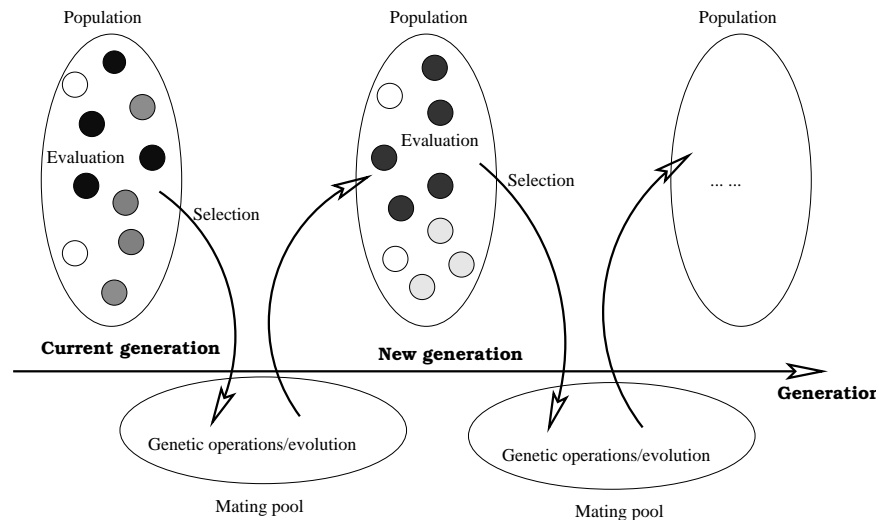
EC Key Idea

- 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

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



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; and 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 fitness, the better the performance
 - the smaller the fitness, the better the performance

Evolutionary Search

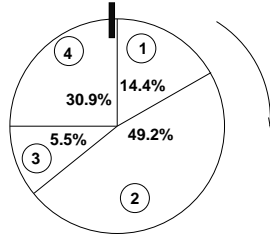
- The current generation
 - A population of candidate solutions
 - Evaluation
- Evaluation: Fitness function/evaluation
 - Performance measure of candidate solutions
 - competition
- Selection: Population → Mating pool for evolution
 - Select good candidates
 - selection pressure
- Evolution: Genetic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

Representations: Genetic Algorithms

- A large branch 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

- 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

Tackling A Problem with GAs

- Formulate the problem as find min/max of $f(p_1, p_2, p_3, \dots, p_n)$
- Ensure that f can be evaluated for all values of p_i
- Encode the 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

Genetic Algorithms: Genetic Operators

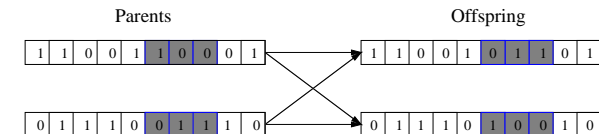
• Crossover

- One point crossover

Parent1: 0 1 1 1 | 1 Child1: 0 1 1 1 0

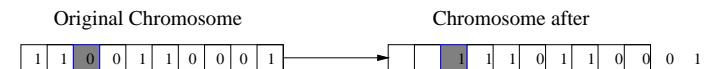
Parent2: 1 1 0 0 | 0 Child2: 1 1 0 0 1

- Two point crossover: for long chromosomes



• Mutation

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



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**