COMP3411: Artificial Intelligence

Extension 5. Evolutionary Computation

Outline

- Evolutionary Computation
- Representations
- Fitness Functions
- Deceptive Landscapes
- Coevolution

Evolutionary Computation

- use principles of natural selection to evolve a computational mechanism which performs well at a specified task.
- start with randomly initialized population
- repeated cycles of:
  - evaluation
  - selection
  - reproduction + mutation
- any computational paradigm can be used, with appropriately defined reproduction and mutation operators

Evolutionary Computation

For concreteness, let's assume we have a population of 100 individuals.

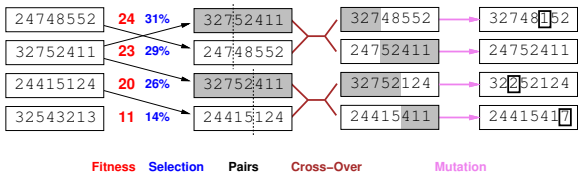
At each generation, we evaluate a fitness score for each individual. In some cases, this may require tranlating from a genotype to a phenotype.

The best 50 individuals are selected, and the other 50 are "culled" or removed from the population.

Crossover and mutation operators are applied to the selected individuals, producing 50 new individuals to replace those who were culled.

We then evaluate the new population of 100 individuals, and the cycle repeats.

Genetic Algorithms



A schema is a genome pattern in which some values are specified and others are not, e.g.

```
**415***
```


Schema "Theorem"

- implicit parallelism
- fitter schemas increase their representation over time
- schemas combine like "building blocks"

Evolutionary Issues

- Representations
- Mutation operators
- Crossover operators
- Fitness functions

Representations

- continuous parameters (Swefel – "Evolutionary Strategy")
- Bit Strings (Holland – "Genetic Algorithm")
- S-expression trees (Koza – "Genetic Programming")
- Lindenmeyer system (e.g. Sims – "Evolving Virtual Creatures")

Bit String Crossovers

one−point crossover:

11101001000 + 00001010101 → 11101010101, 00001001000

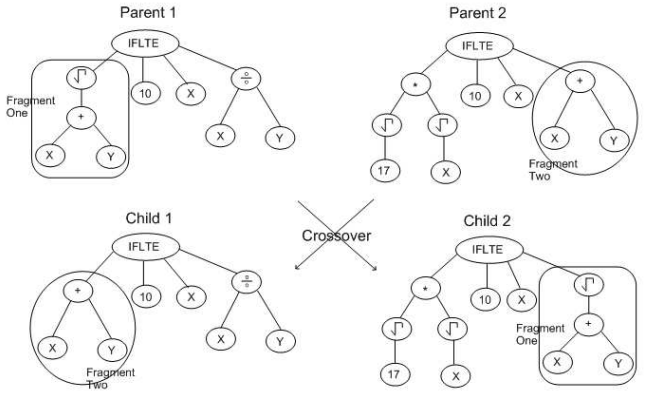
two−point crossover:

11101001000 + 00001010101 → 11001011000, 00101000101

point mutation:

11101001000 → 11101011000

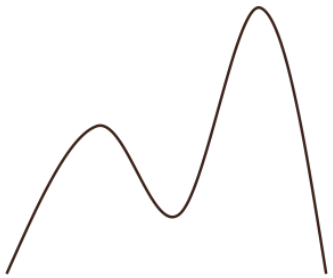
S-expression Trees (Genetic Programming)



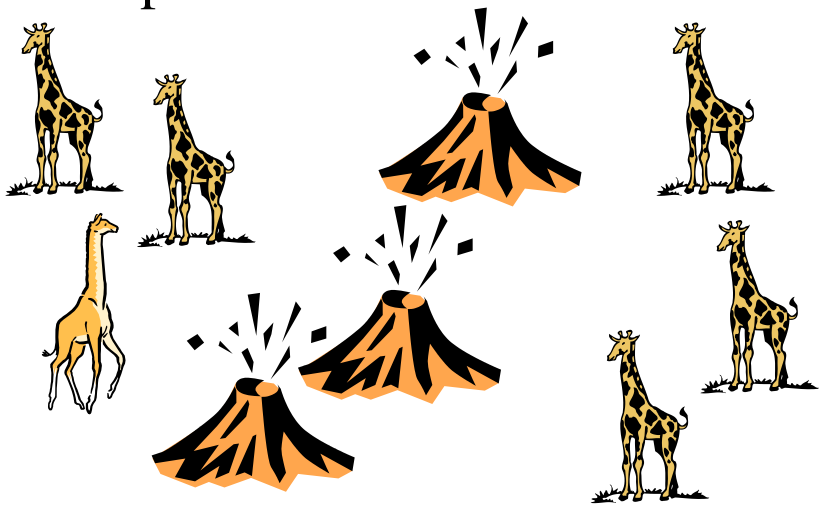
Fitness Functions

Sometimes the fitness function present as sooth "hill" for the algorithm to climb.
But, often we see "deceptive" landscapes leading to premature convergence, where the population get stuck on a local opmimum.

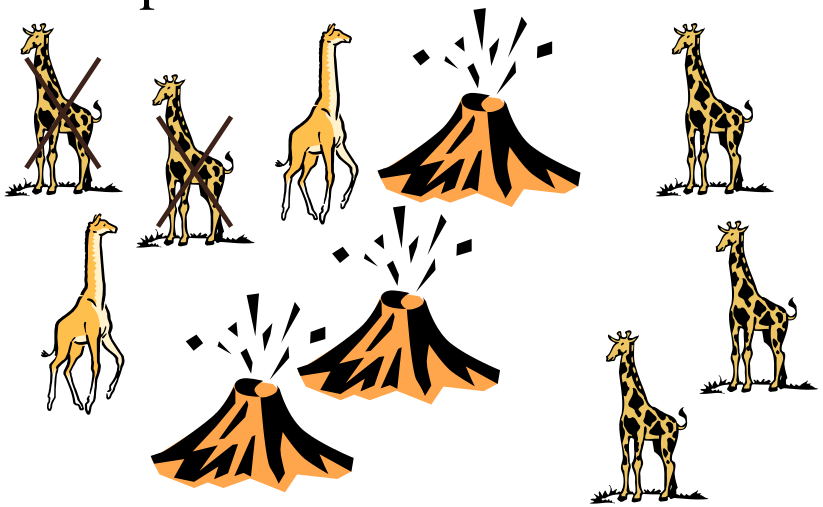
- fitness sharing
- random re-starts
- Age Layered Planes
- (spatial) coevolution



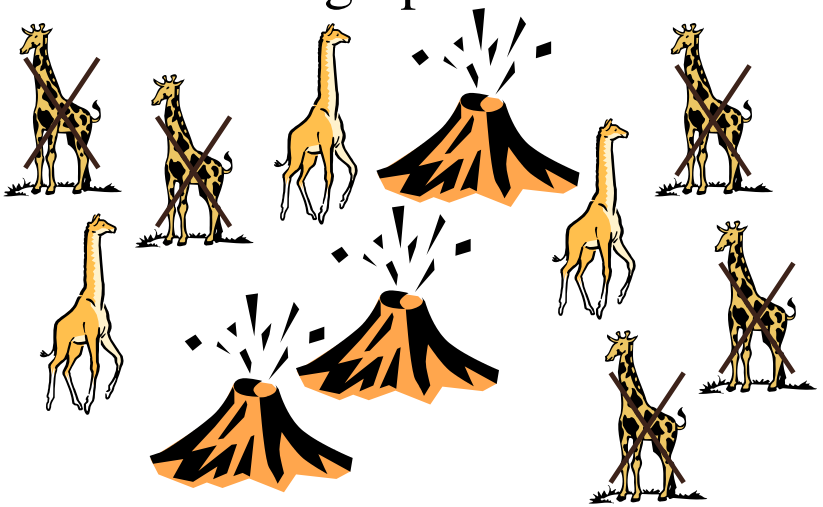
"Gaps" in the Fossil Record?



“Gaps” in the Fossil Record?



Partial Geographic Isolation



Punctuated Equilibria

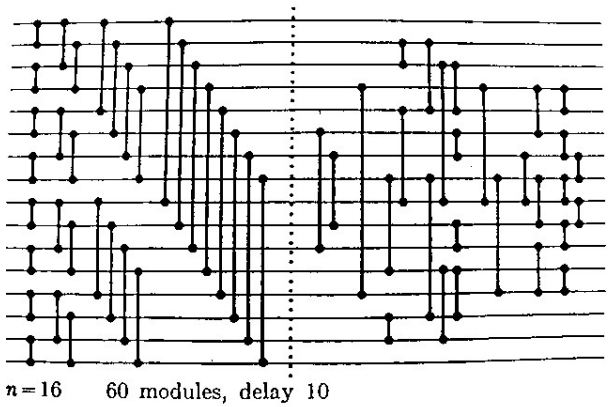
“Gaps” in the Fossil Record?

- Eldridge & Gould
  - partial geographic isolation
  - punctuated equilibria
- ideas for Evolutionary Computation?
  - “island” models
  - co-evolution / artificial ecology ?

Co-Evolution

- competitive (leapard vs. gazelle)
- co-operative (insects/flowers)
- mixed co-operative/competitive (Maynard-Smith)
- different genes within same genome?
- "diffuse" co-evolution

Sorting Networks



Sorting Networks #1 (Hillis)

- Evolving population of networks
- converged to local optimum
- final network not quite as good as hand-crafted human solution

Sorting Networks #2 (Hillis)

- two co-evolving populations (networks and strings)
- can escape from local optima
- punctuated equilibria observed
- better than hand-crafted solution (Tufts, Juillé & Pollack)

Co-evolutionary Paradigms

- machine vs. machine (Sims)
- human vs. machine (Tron)
- mixed co-operative/competitive (IPD)
- brain / body (Sims, Lipson)
- language games (Tonkes, Ficici)
- single individual ? (Backgammon)
- Generative Adversarial Networks

Iterated Prisoner's Dilemma

	C	D
C	3, 3	0, 5
D	5, 0	1, 1

- TFT ⇨ ALL-C ⇨ ALL-D ⇨ TFT