COMP9444
Neural Networks and Deep Learning

13. Coevolution

Outline

- Evolutionary Computation Paradigms
- Deceptive Landscapes
- Punctuated Equilibria
- Coevolution in Nature
- Coevolution in Machine Learning

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

Recall: Hill Climbing

- Initialize "champ" policy $\theta_{\text{champ}} = 0$
- for each trial, generate "mutant" policy

$$\theta_{\text{mutant}} = \theta_{\text{champ}} + \text{Gaussian noise (fixed } \sigma)$$

- champ and mutant play a number of games, with same game initial conditions
- if mutant does "better" than champ,

$$\theta_{\text{champ}} \leftarrow (1-\alpha)\theta_{\text{champ}} + \alpha\,\theta_{\text{mutant}}$$

We saw this algorithm applied to Backgammon, and Simulated Hockey.

Evolutionary Computation

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.

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:

$\underline{11101}001000 \quad \rightarrow \quad 11101010101$

$00001\underline{010101} \quad \rightarrow \quad 00001001000$

two−point crossover:

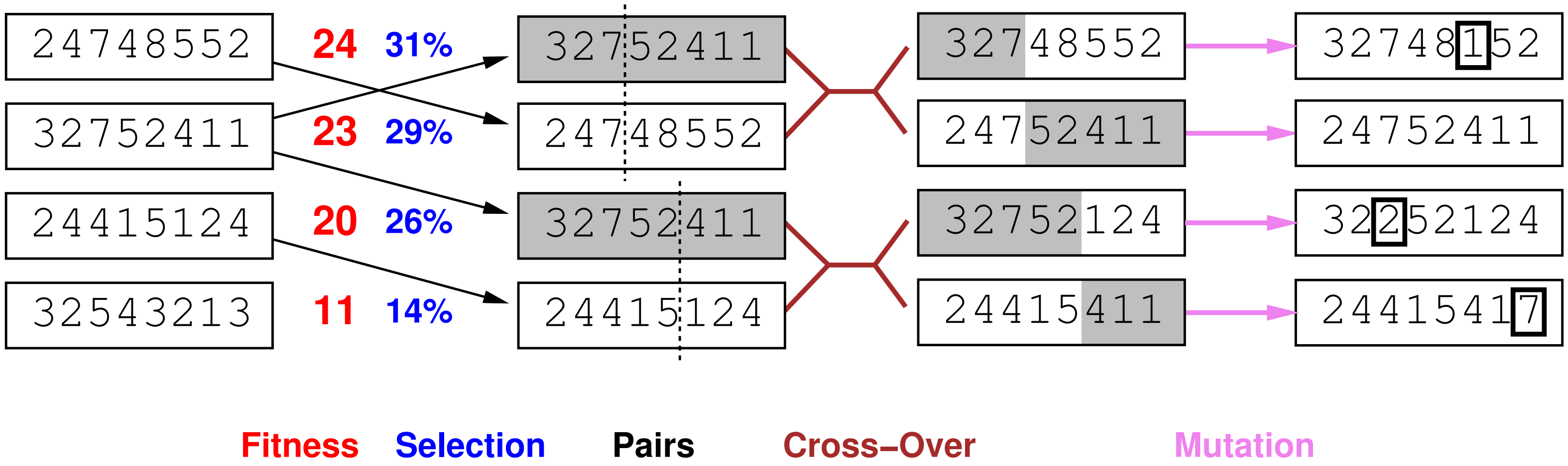
$11\underline{10100}1000 \quad \rightarrow \quad 11001011000$

$\underline{00}00101\underline{0101} \quad \rightarrow \quad 00101000101$

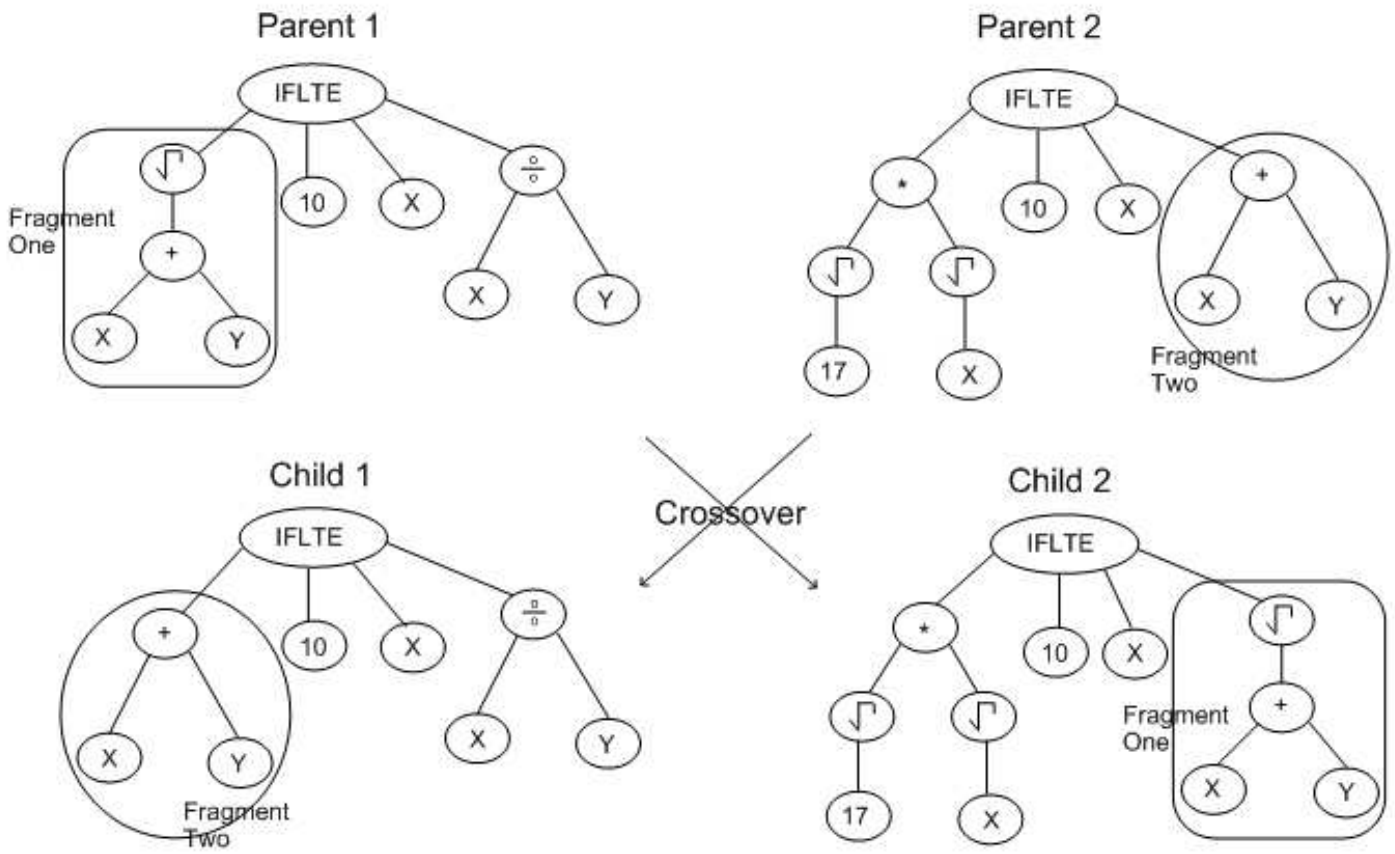
point mutation:

$111010\underline{0}1000 \longrightarrow 11101011000$

Genetic Algorithms



S-expression Trees (Genetic Programming)



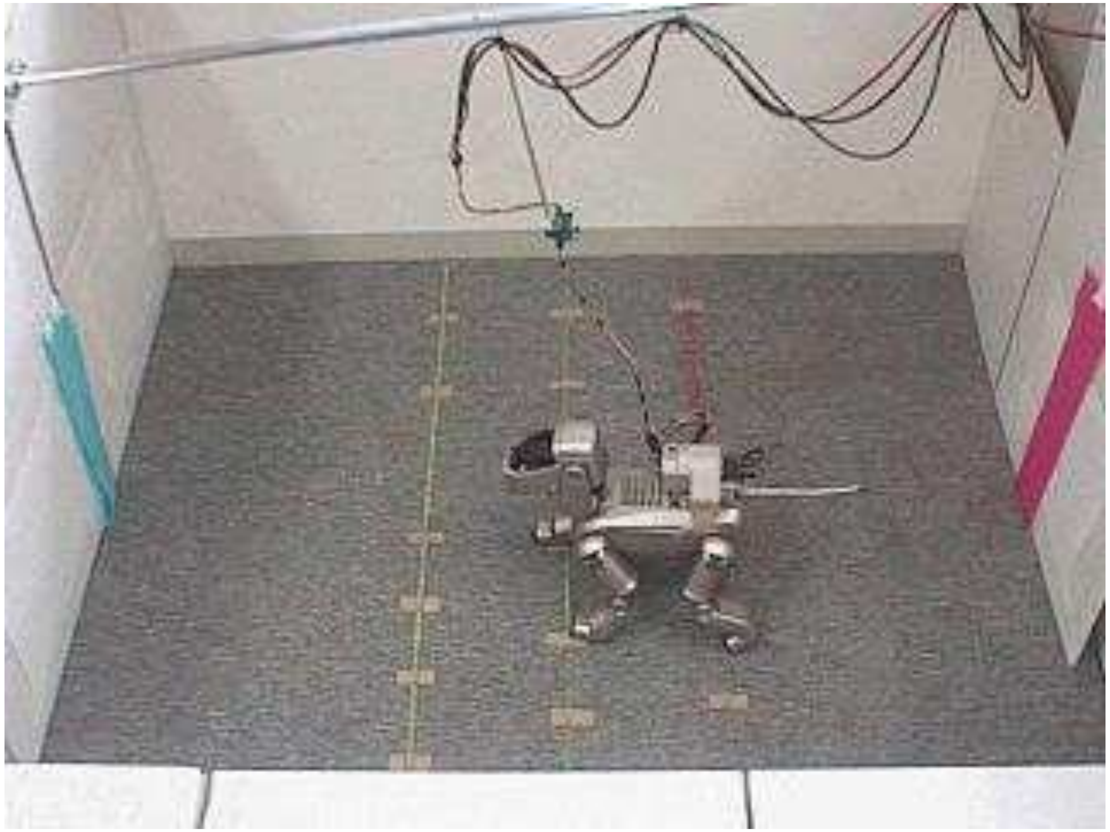
Fitness Functions

Sometimes the fitness function presents a sooth "hill" for the algorithm to climb.

But, often we see "deceptive" landscapes leading to premature convergence, where the population gets stuck on a local opmimum.

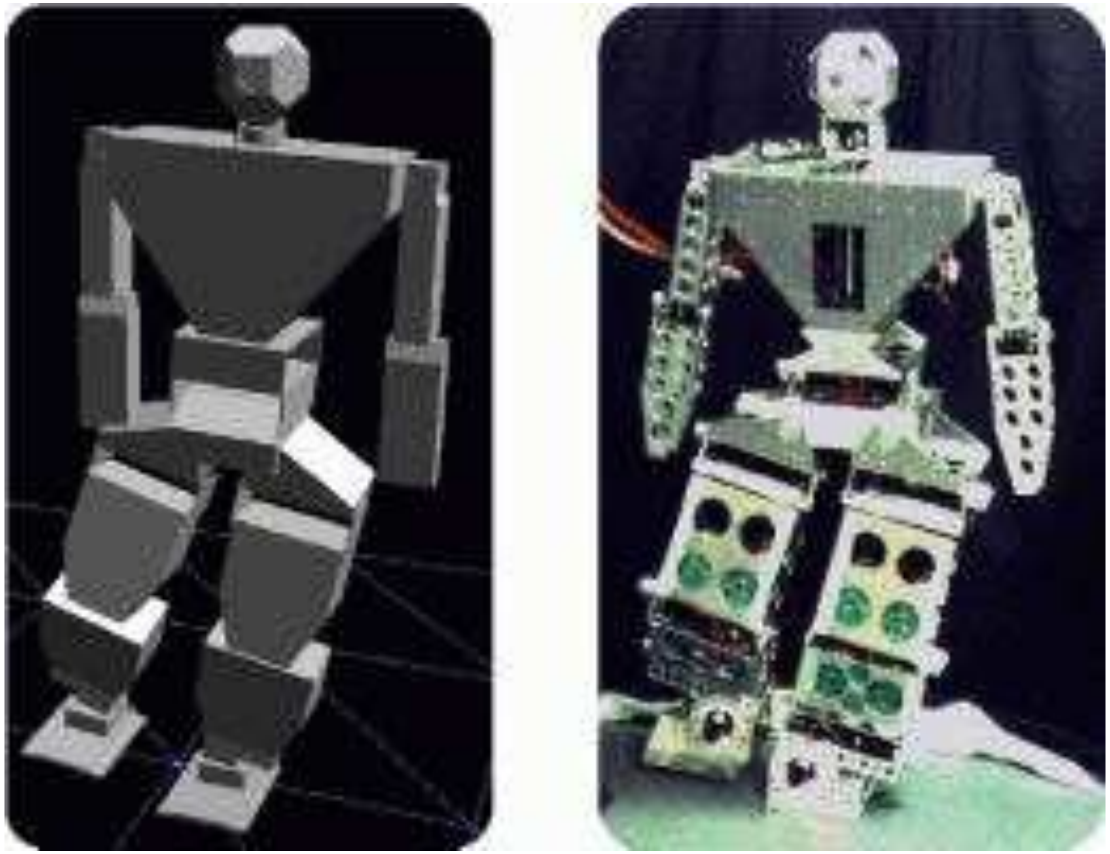
- fitness sharing
- random re-starts
- age layered planes (ALPS)
- (spatial) coevolution

Aibo Walk Learning (Hornby)



- Learning done on actual robot.

Guroo – Humanoid Walk Learning



- Learning done in simulator(s), then tested on actual robot.

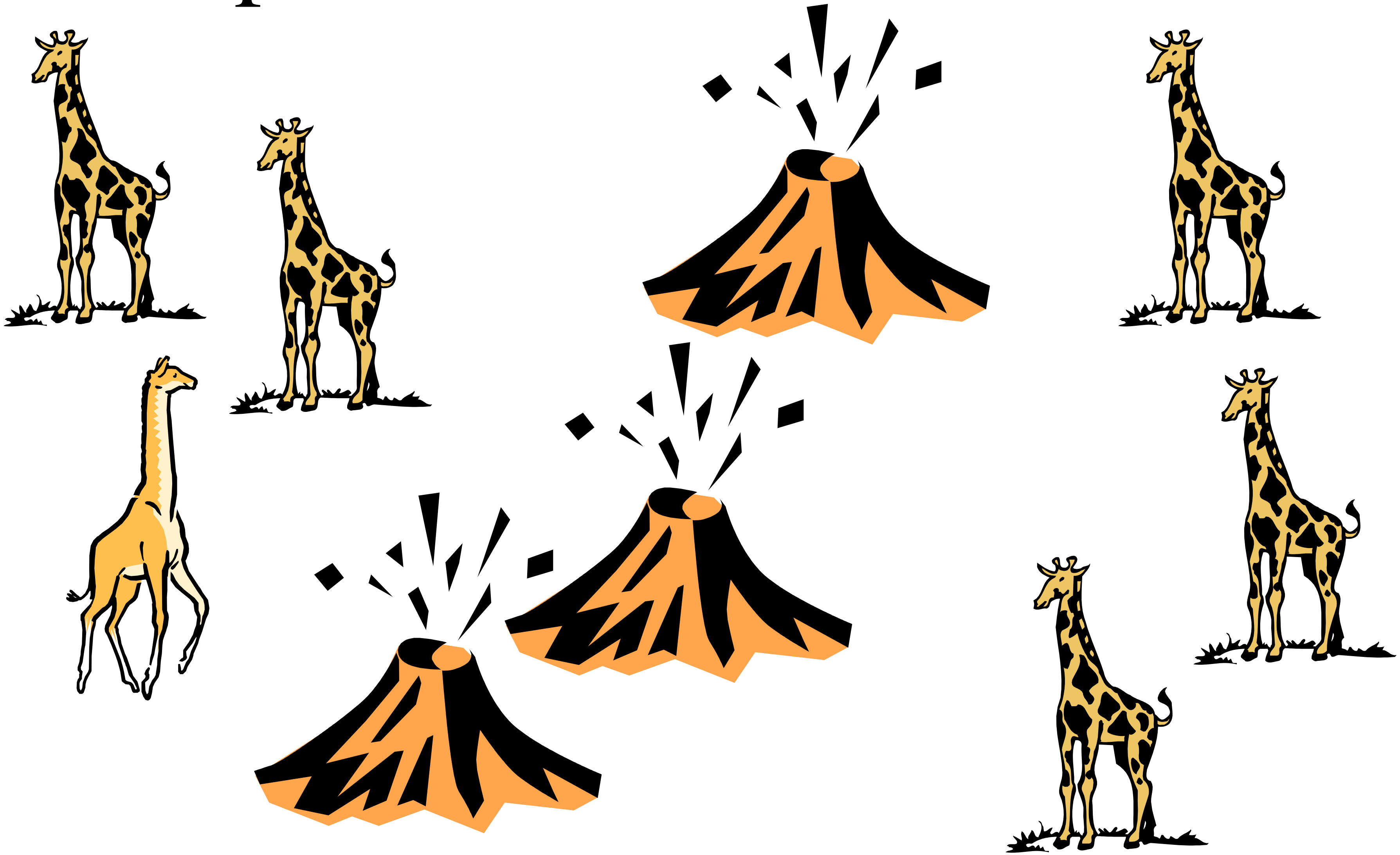
Evolved Antenna

One example of the use of Evolutionary Algorithms for a real world application is the antenna that was evolved by Hornby et al in 2006 for NASA's Space Technology 5 (ST5) mission.

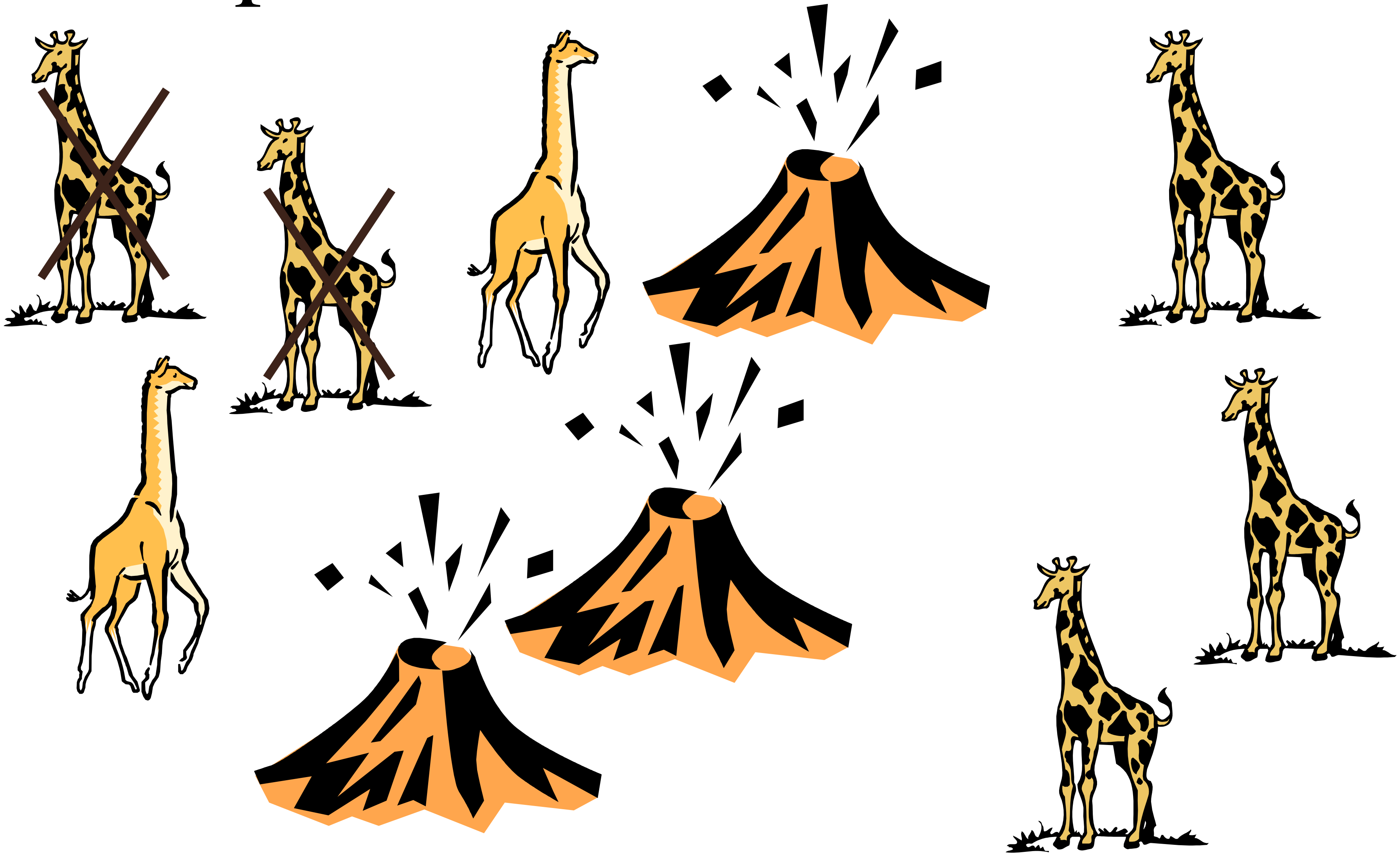
“Gaps” in the Fossil Record?

- Eldridge & Gould, 1970
  - partial geographic isolation
  - punctuated equilibria
- ideas for Evolutionary Computation?
  - “island” models
  - co-evolution / artificial ecology ?

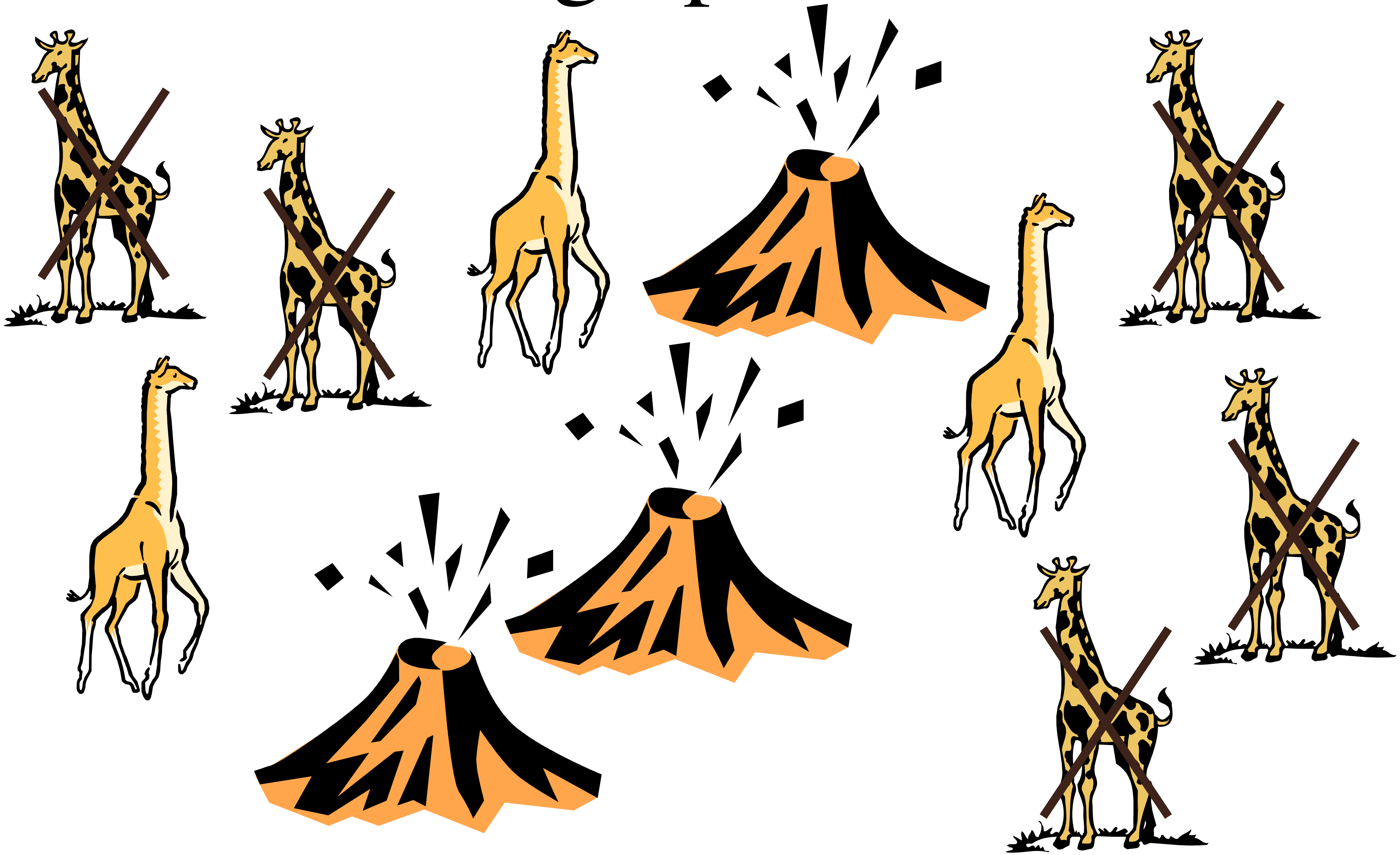
“Gaps” in the Fossil Record?



"Gaps" in the Fossil Record?



Partial Geographic Isolation

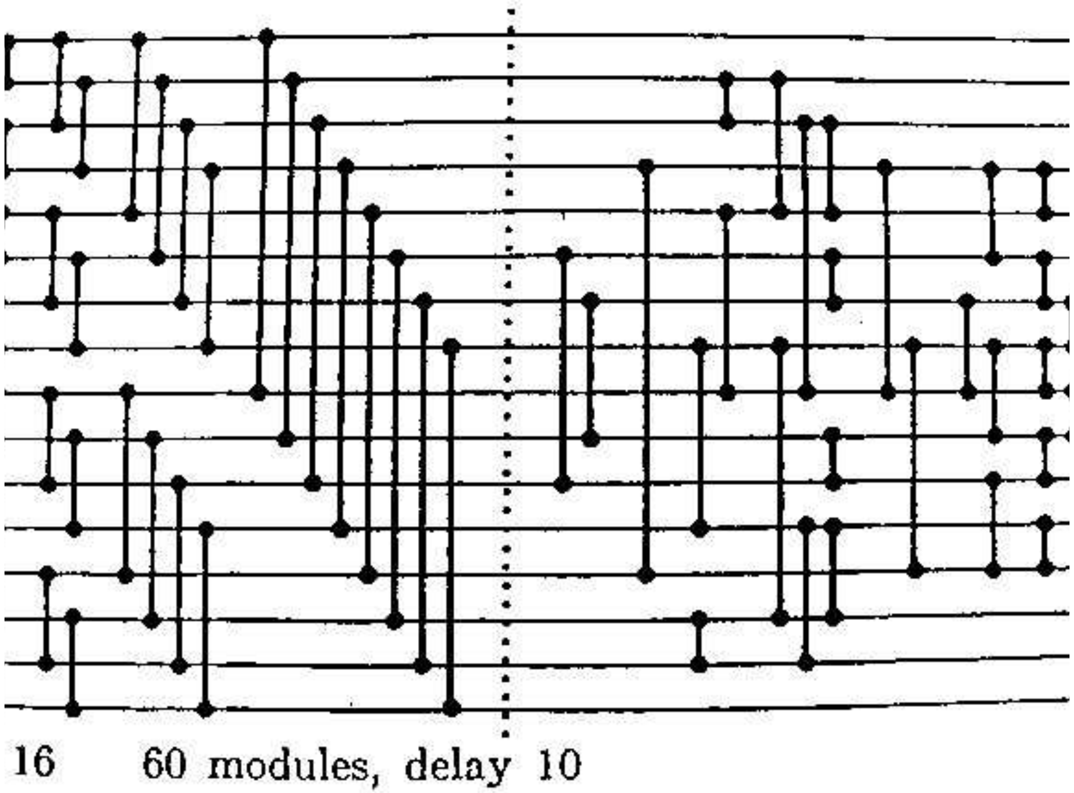


Punctuated Equilibria

Co-Evolution in Nature

- competitive (leopard vs. gazelle)
- co-operative (insects/flowers)
- mixed co-operative/competitive (Maynard-Smith)
- different genes within the 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

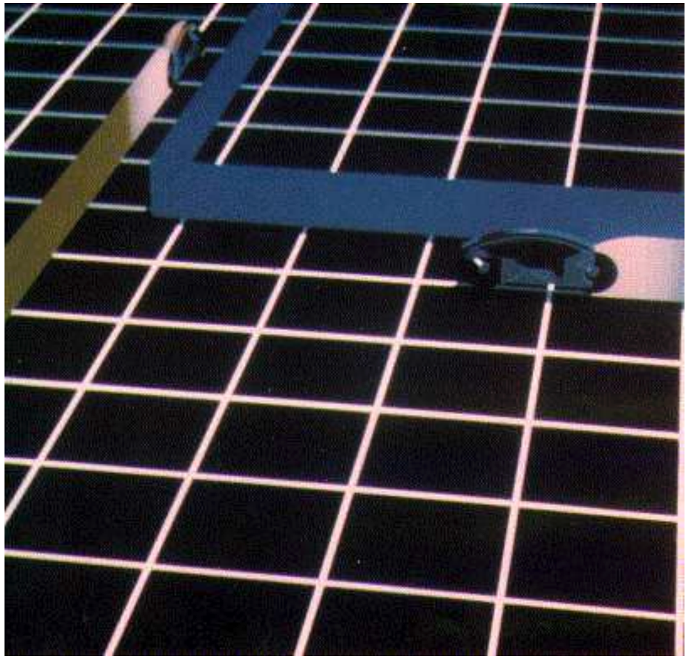
Sorting Networks #2 (Hillis)

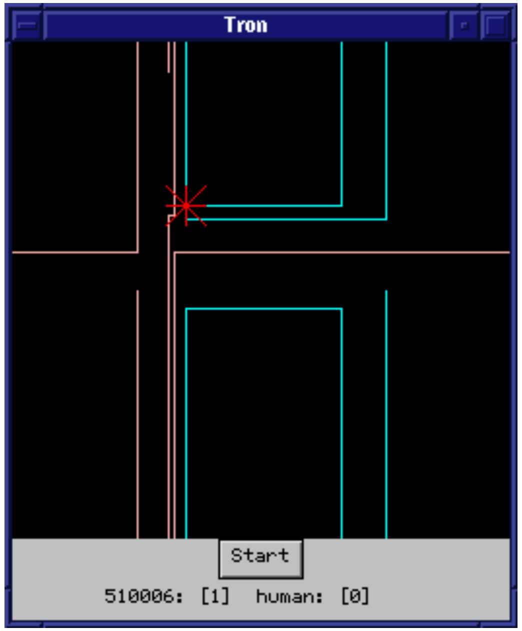
- two co-evolving populations (networks and strings)
- can escape from local optima
- punctuated equilibria observed
- better than hand-crafted solutions (Tufts, Juillé & Pollack)

Co-evolution in Machine Learning

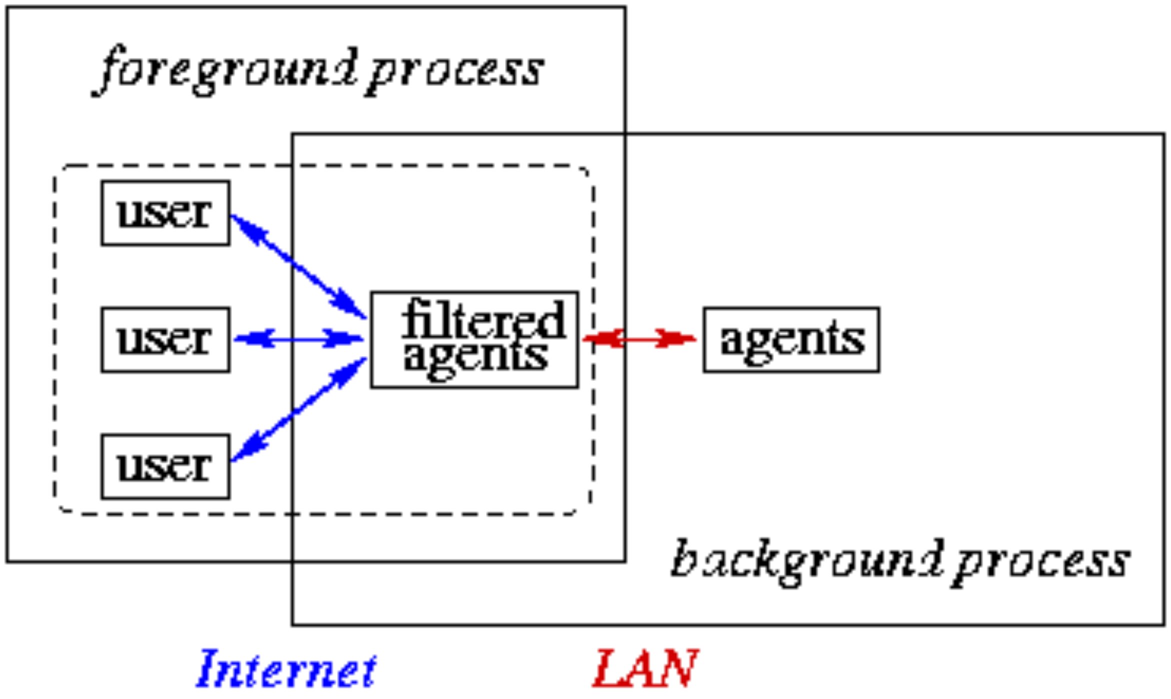
- machine vs. machine (Hillis)
- human vs. machine (Tron)
- mixed co-operative/competitive (IPD)
- brain / body (Sims, Lipson)
- language games (Tonkes, Ficici)
- single individual ? (Backgammon)
- Generative Adversarial Networks

Tron





Tron


foreground process
user
user
user
filtered agents
agents
background process
Internet
LAN

Iterated Prisoner's Dilemma

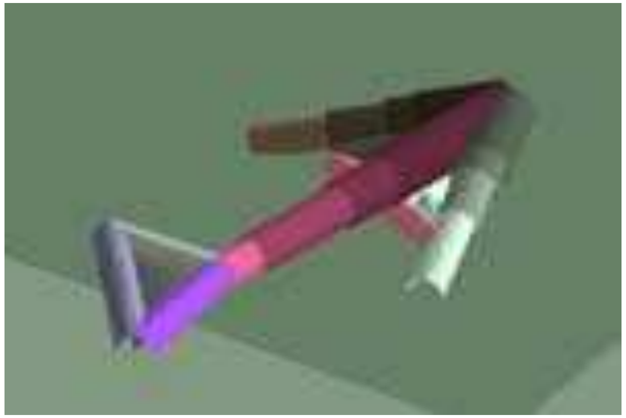
	C	D
C	3,3	0,5
D	5,0	1,1

TFT $\rightarrow$ ALL-C $\rightarrow$ ALL-D $\rightarrow$ TFT

Evolving Virtual Creatures (Sims)

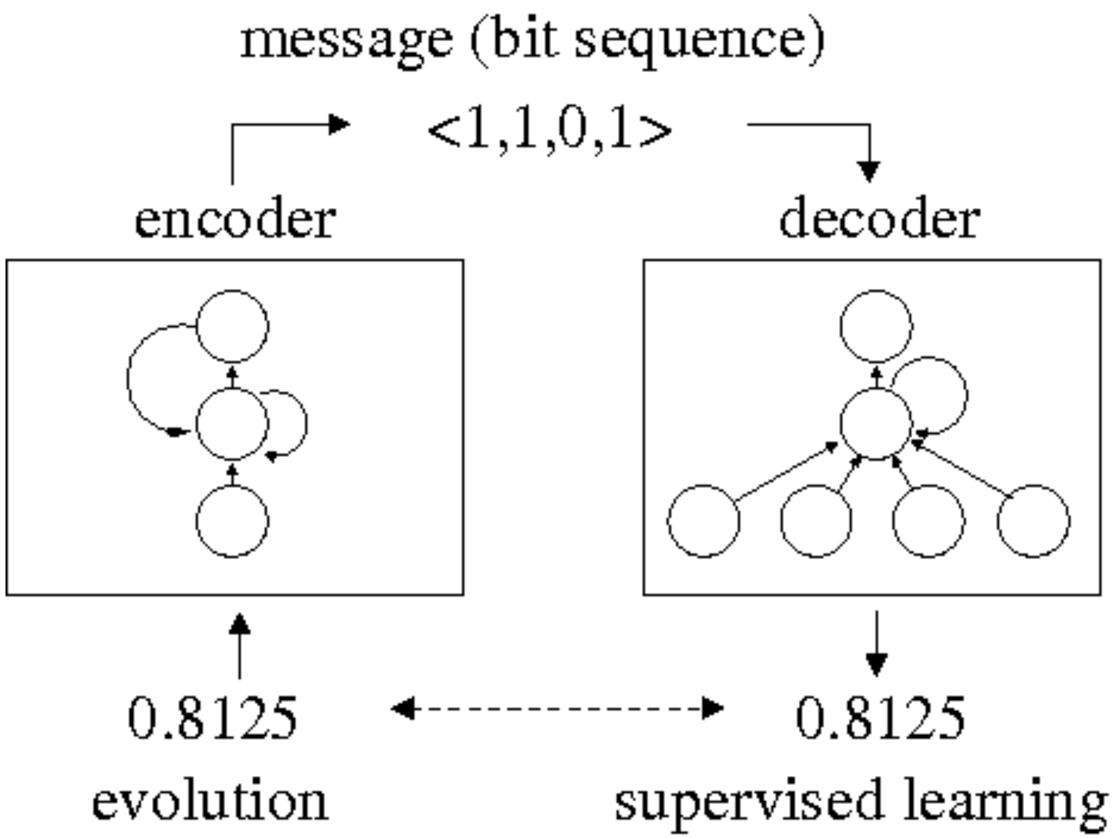
- Body evolves as a Lindenmeyer system
- Controller evolves as a neural network

Golem (Lipson)

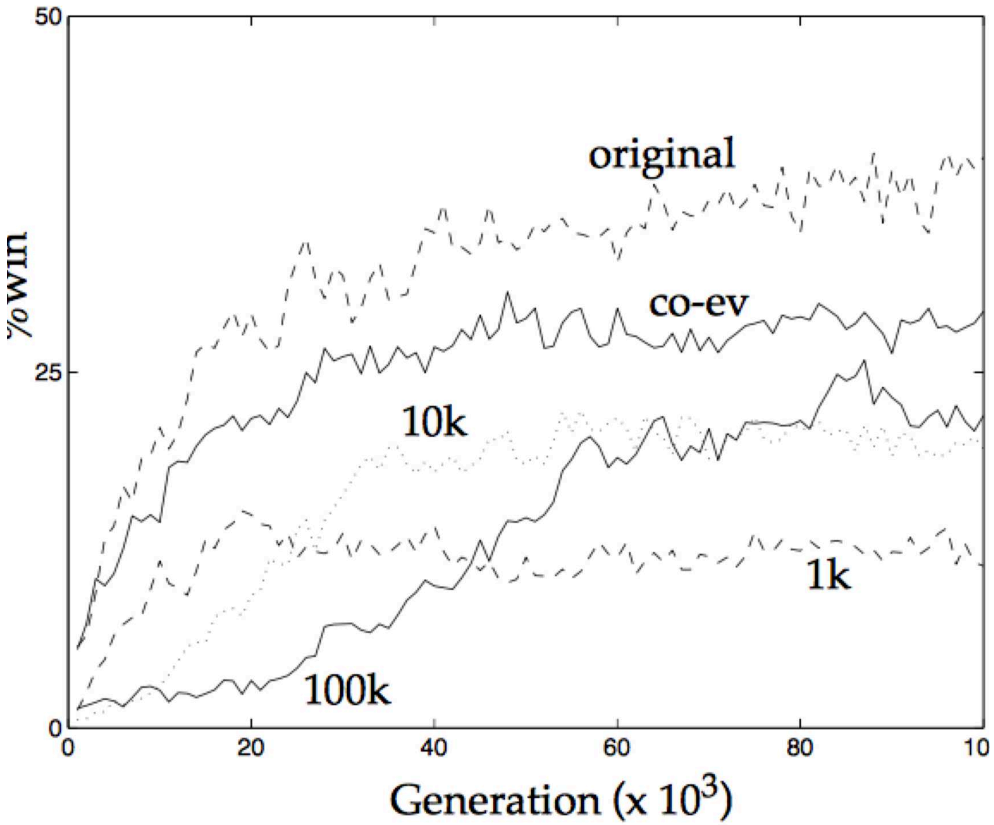


- Evolved in simulation, tested in reality.

Language Games

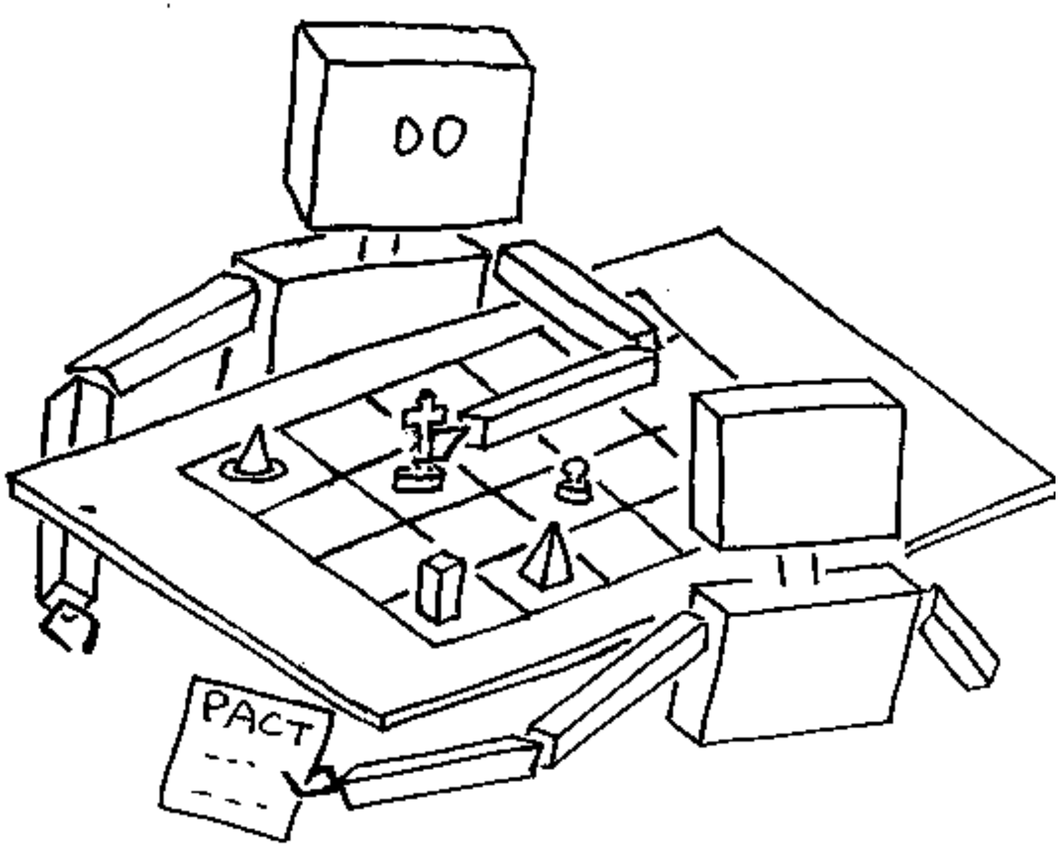


Self-Play as Coevolution



HC-Gammon trained by self-play, and against fixed opponents.

Collusion



Meta-Game of Learning

- Co-evolution tends to provide an opponent of appropriate ability
- generally helps to escape from local optima
- however, can create new “mediocre stable states” (collusion)