Evolving Neural Networks through Augmenting Topologies

Authors

Kenneth O. Stanley

Risto Miikkulainen

Speaker

Daniele Loiacono

ARTIFICIAL INTELLIGENCE and ROBOTICS LABORATORY

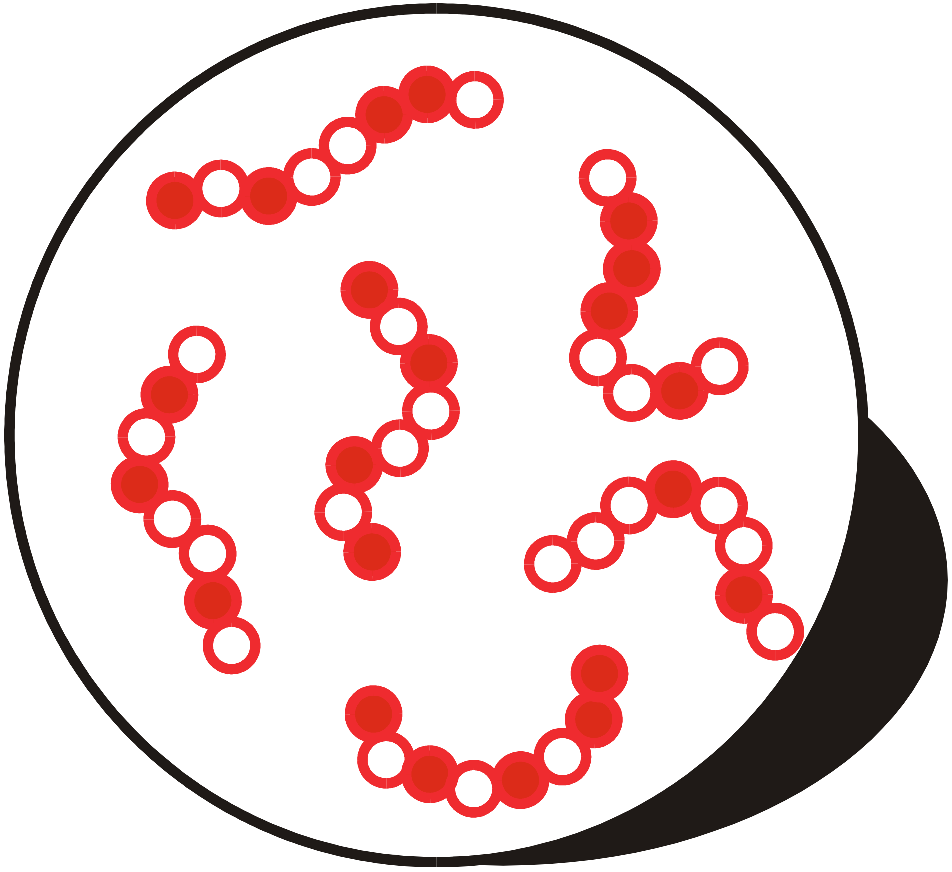
Outline

- Introduction to Genetic Algorithms
- Introduction to Neural Networks
- Neuroevolution
- NEAT

Outline

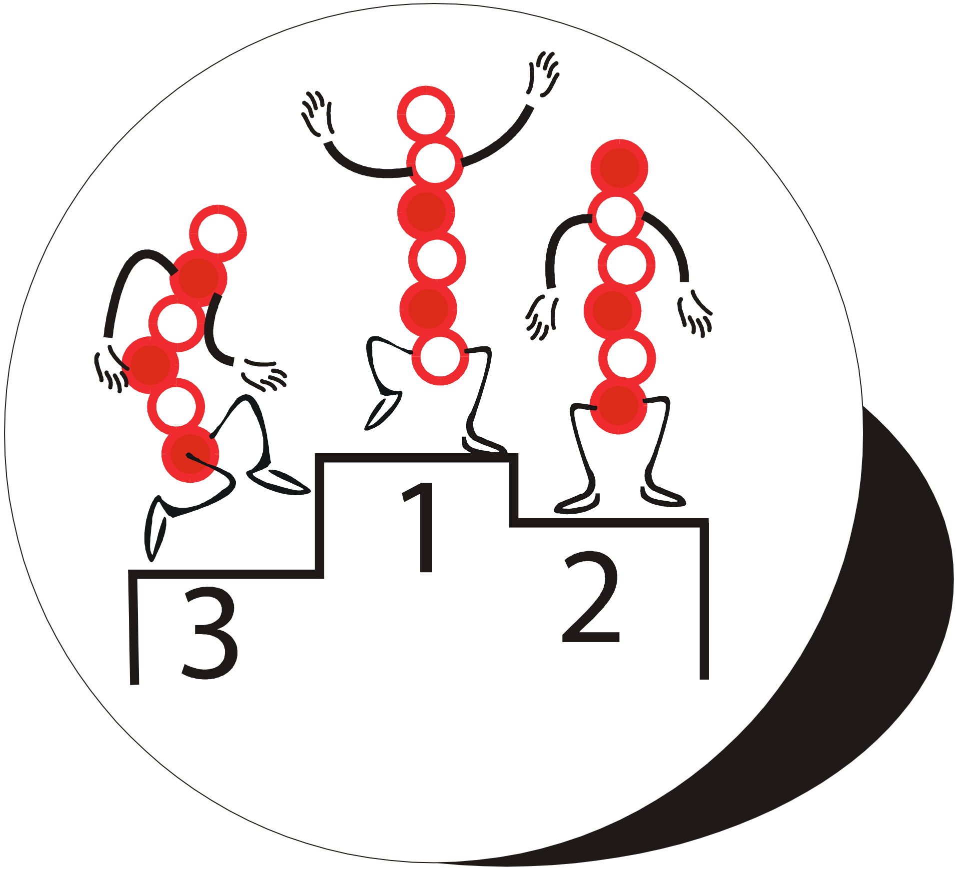
- Introduction to Genetic Algorithms
- Introduction to Neural Networks
- Neuroevolution
- NEAT

The population



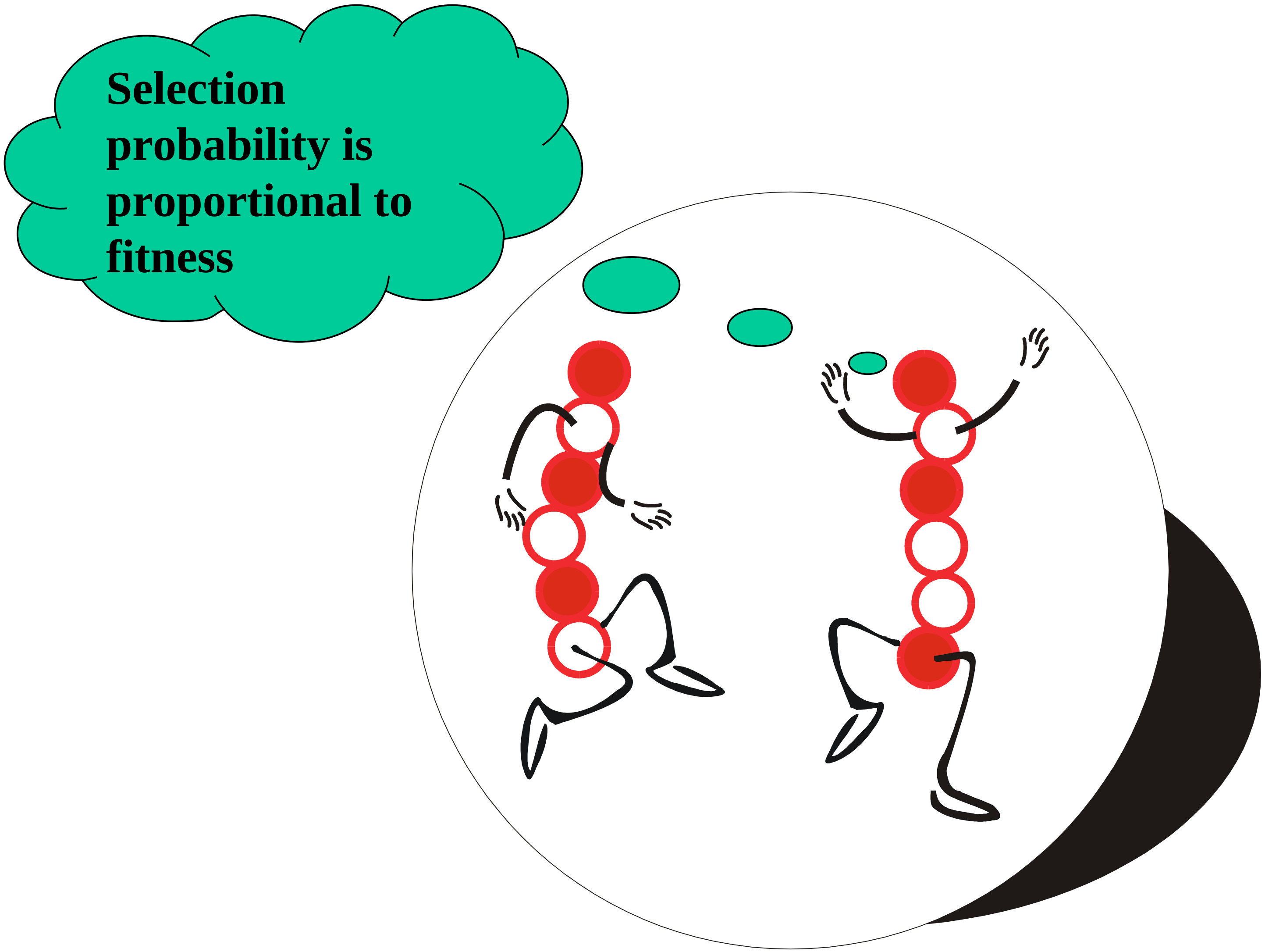
David Brogan, 2006

Ranking by fitness



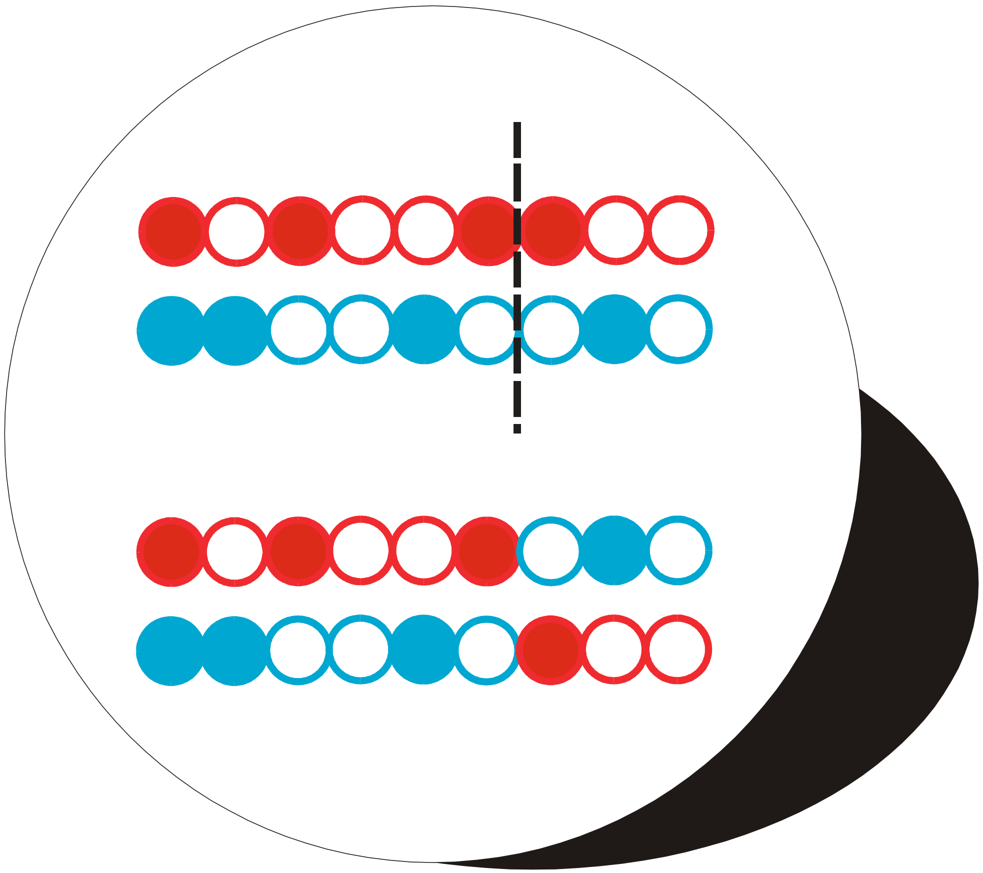
David Brogan, 2006

Mate Selection



David Brogan, 2006

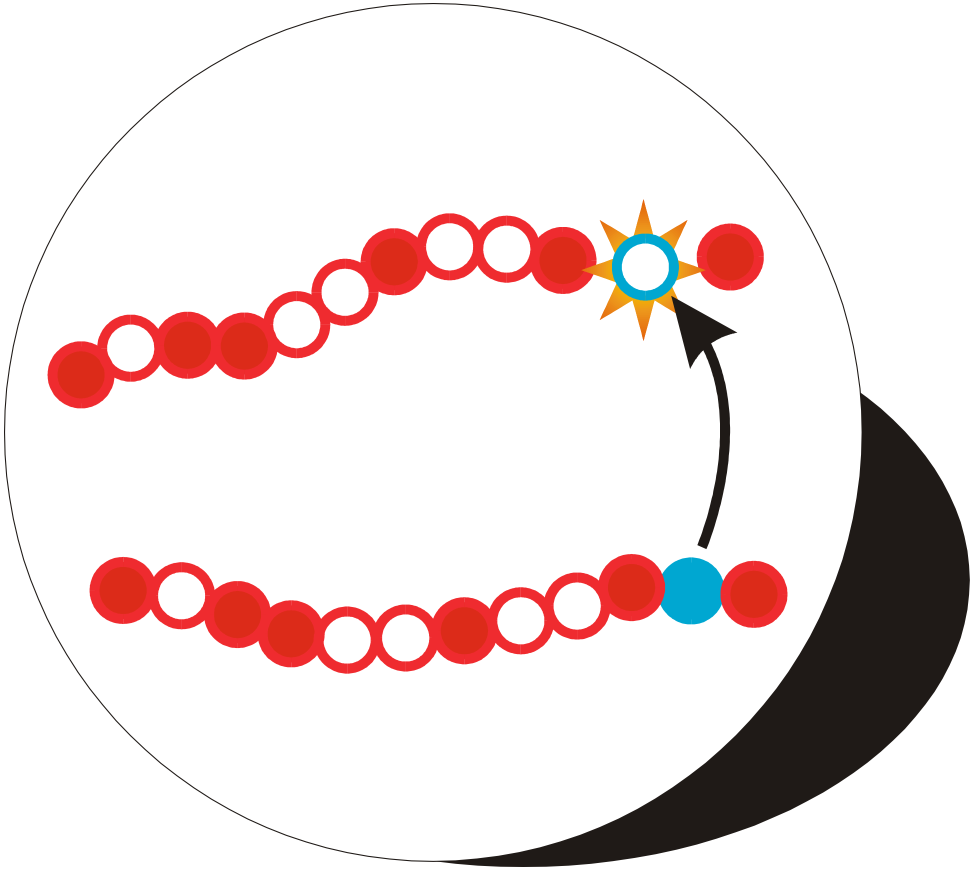
Exploit goodness
of
parents



David Brogan, 2006

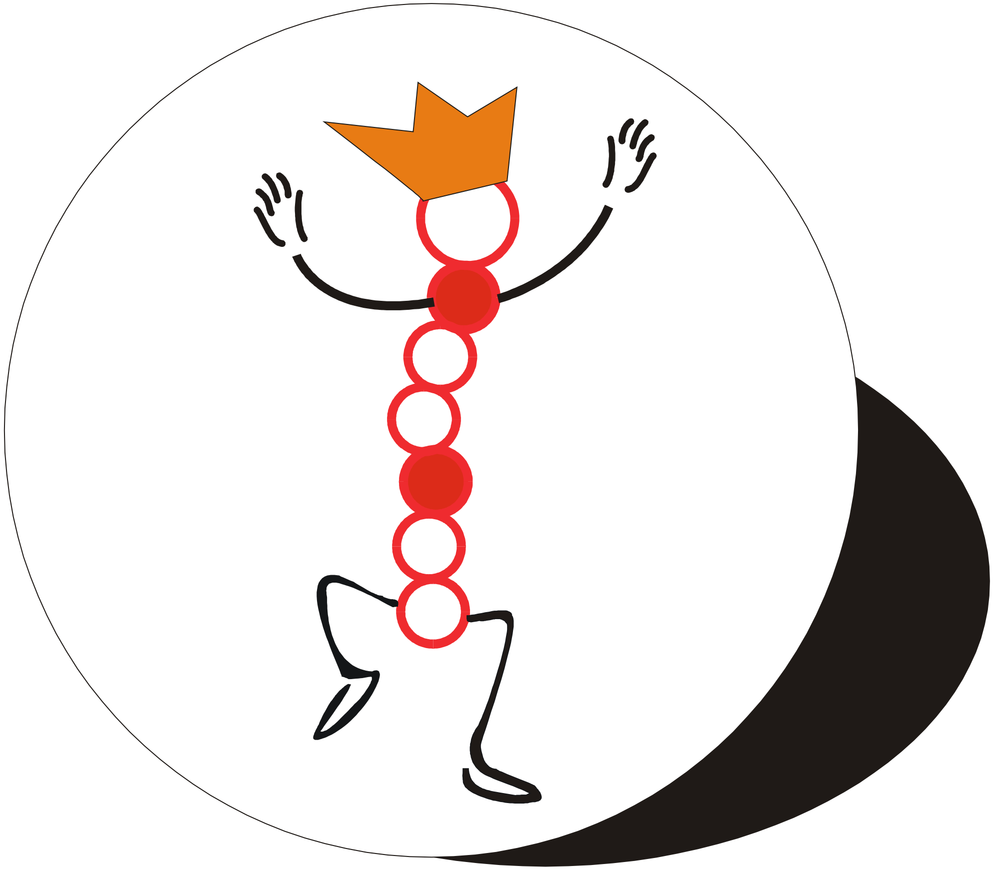
Mutation

Explore unknown

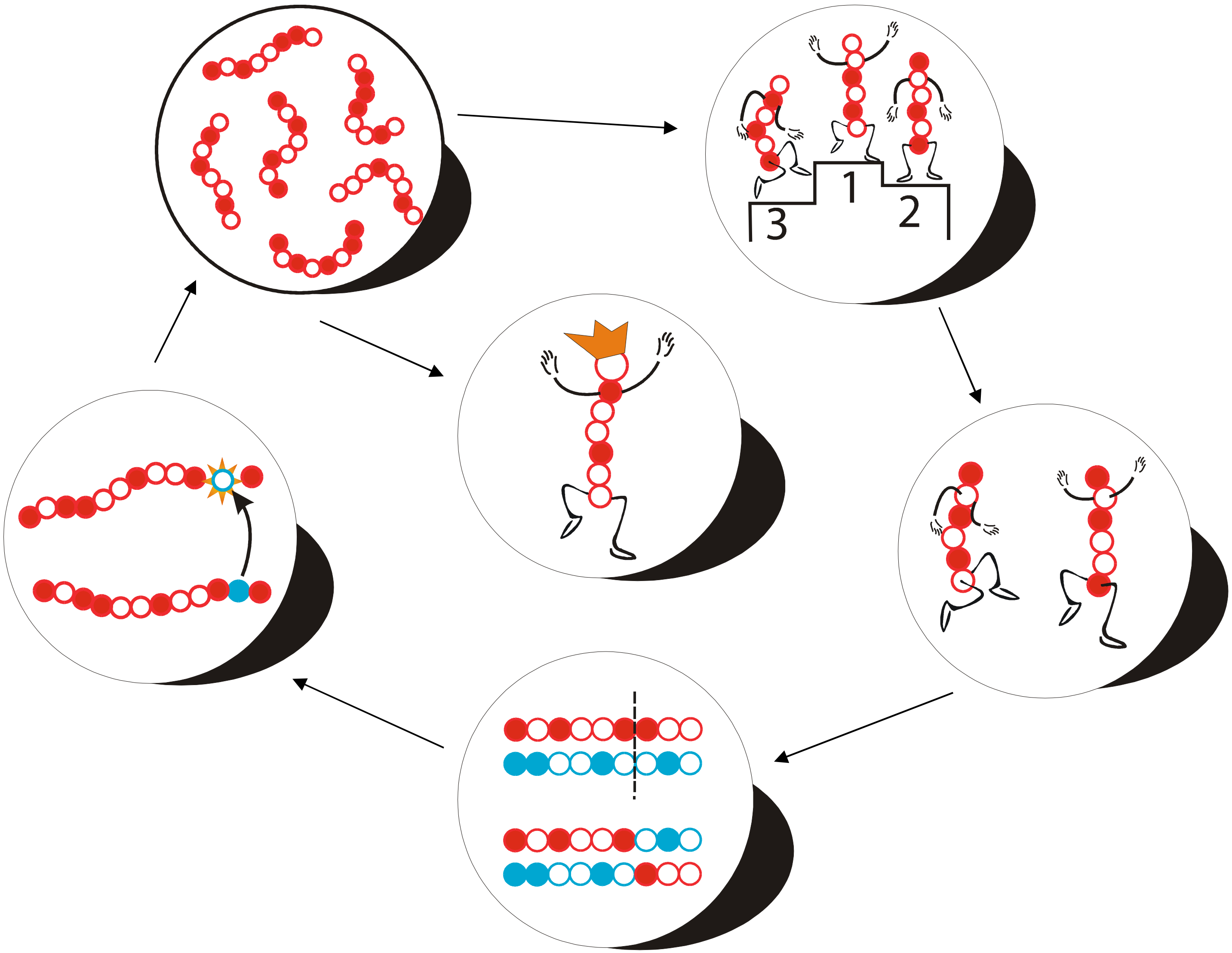


David Brogan, 2006

Best solution



David Brogan, 2006



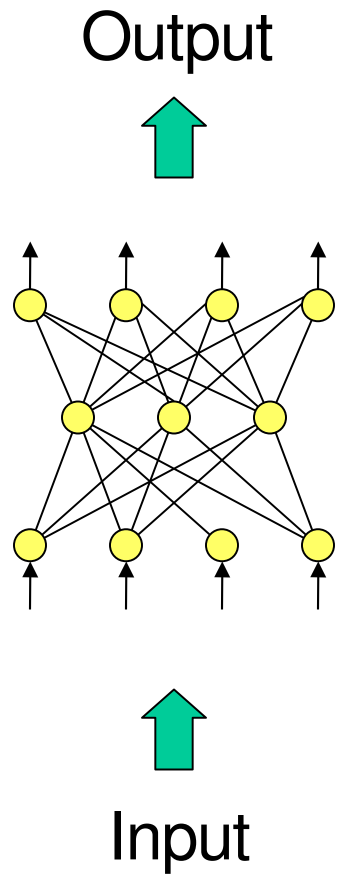
David Brogan, 2006

Outline

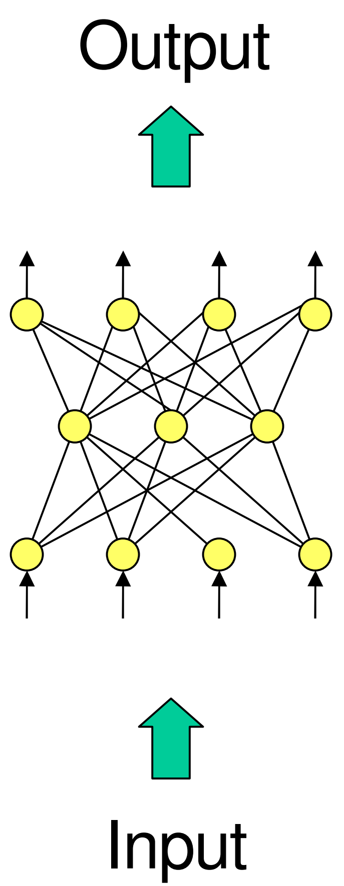
- Introduction to Genetic Algorithms
- Introduction to Neural Networks
- Neuroevolution
- NEAT

Neural Networks

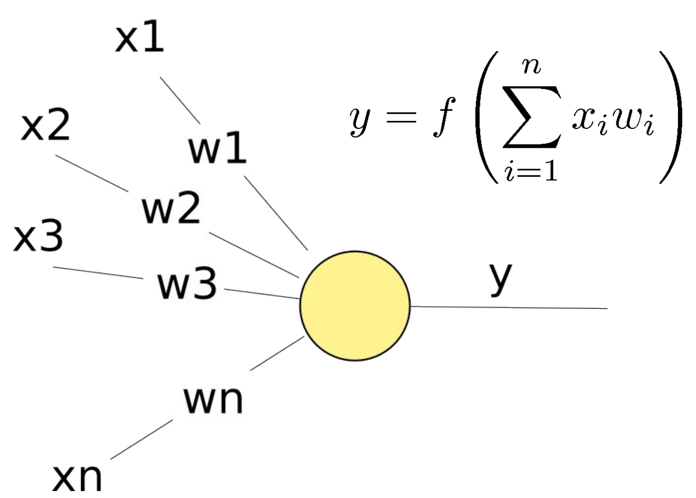
No *much* more than a black box…



Just a glimpse inside the box



- Neurons
- Layers
- Connections and weights

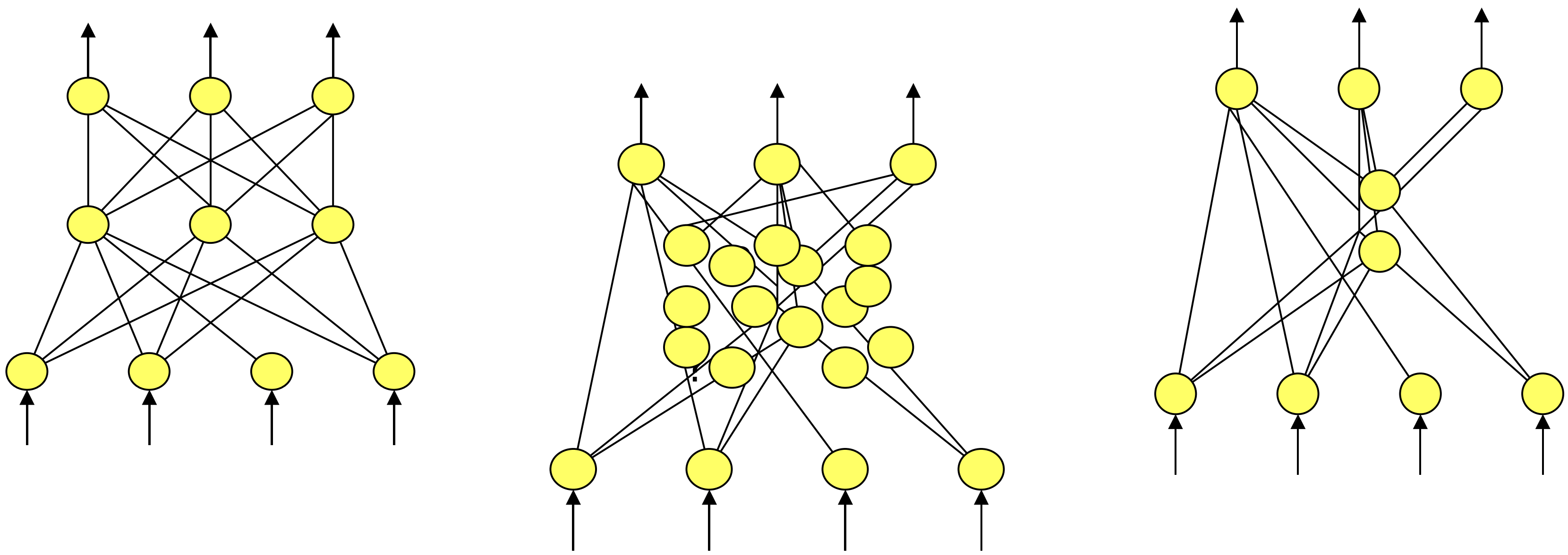


Learning weights of NNs

- Weights of NNs can be learned from a set of samples <x,y>, in order to minimize the **network error**
- Many algorithms have been introduced in the literature (e.g. error backpropagation)
- Unfortunately:
  - we do not have always a set of samples
  - learning can get stuck in local minima

Design of network topology

- What is the best network topology for solving a given problem?
- A key factor for a successful application of NNs!
- Unfortunately is a trial and error process!



Outline

- Introduction to Genetic Algorithms
- Introduction to Neural Networks
- Neuroevolution
- NEAT

Neuroevolution (NE)

- Artificial evolution of NN using GA
- Fitness is the evaluation of NN performance
- Advantages:
  - no need for set of samples
  - avoid local minima
- Evolving what?
  - weights in a fixed network topology
  - topology along with weights (**TWEANN**)

TWEANN issues

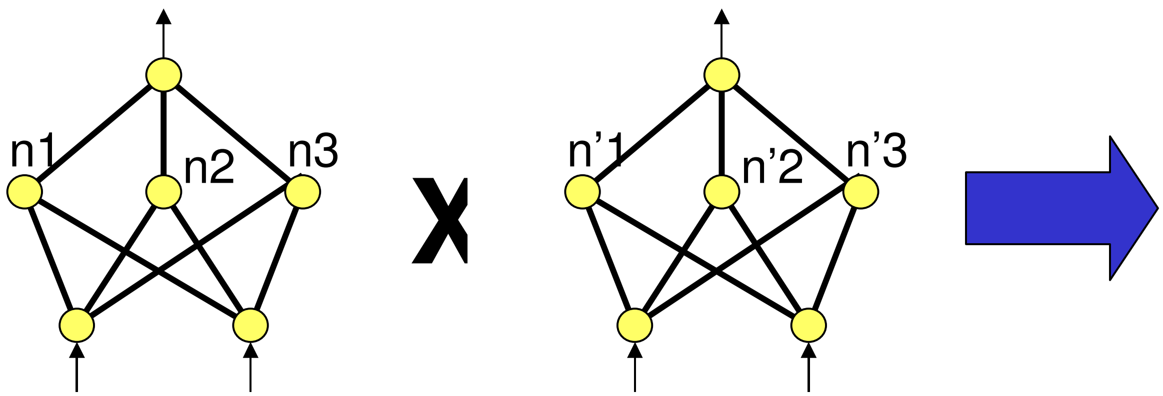
- Encoding
  - direct
  - indirect
- Mating (Crossover)
  - competing conventions
  - free topology and the Holy Grail
- Protecting Innovations
- Initialization and topology minimization
- Examples of TWEANN systems

TWEANN Encoding

- How to encode a neural network?
- Direct encoding
  - genome specifies explicitly the phenotype
  - many encoding proposed (from binary encoding to graph encoding)
- Indirect Encoding
  - genome specify *how to build* the network
  - allow for a more compact representation
  - can be biased

Mating in TWEANN

- How can we mate two networks in meaningful way?
- Challenging even for small networks with the same topology:



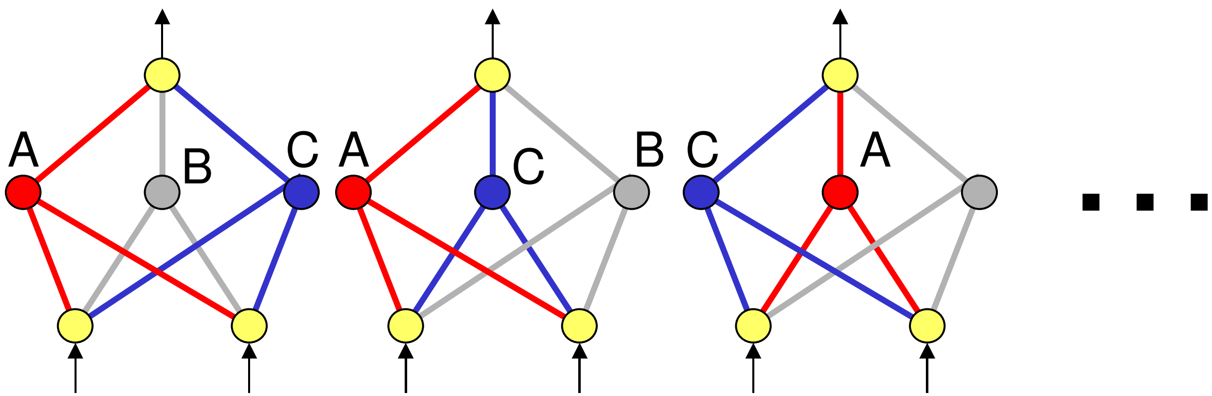
n1 ↔ n'1

n2 ↔ n'2

n3 ↔ n'3

Mating in TWEANN

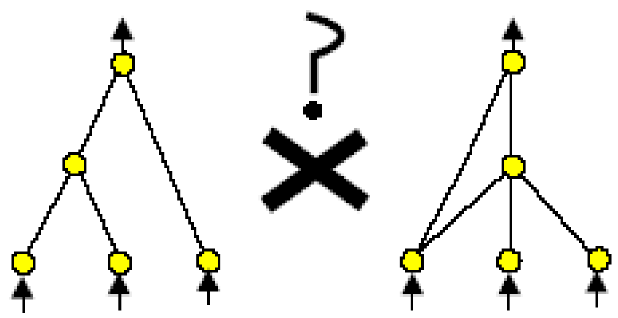
- How can we mate two networks in meaningful way?
- Challenging even for small networks with the same topology:



Competing Conventions Problem

Mating in TWEANN

- For fixed and constrained topologies the competing conventions problem can be solved with tailored encoding
- But with free topology?



The Holy Grail (Radcliffe, 1993)

Protecting innovations

- Innovations usually involves larger networks with more connections
- Larger networks require more time to be optimized and cannot compete with smaller ones (at least in the short run)
- **Speciation** is an effective technique to avoid competition and mating between networks too different

Initialization and topology minimization

- Initial population of random networks
  - networks not connected
  - heterogeneous networks to mate
- How can network topology be minimized?
  - networks bigger than necessary are expensive to optimize
  - fitness penalty

Structured Genetic Algorithm (sGA) Dasgupta and McGregor (1992)

- Binary encoding: a bit string represents the connective matrix of networks
- Advantages
  - Evolution can be performed with a standard GA
- Drawbacks
  - Computational space is square w.r.t. the number of nodes
  - Number of nodes limited

Breeder Genetic Programming (Zhang and Muhlenbein)

- Encoding with tree
- Only crossover adapts topology
- Fitness has an explicit penalty term for bigger networks
- How do we set the penalty? Does it depend from the problem?

Parallel Distributed Genetic Programming (PDGP)
Pujol and Poli (1997)

- Dual encoding:
  - linear genome for mutating weights
  - Graph representation for a subgraph swapping crossover
- Does swapping crossover guarantee a good mating?

- Graph Encoding
- Avoid mating because considered disruptive

- Indirect encoding (developmental)
- Biased search in topology space
- Mating results are unpredictable
- More compact representation
- Effective but not always more effective than direct encoding

Enforced SubPopulations (ESP)
Gomez and Miikkulainen (1997,1999)

- Fixed topology
- When it fails starts from scratch with a new random topology
- Faster than Cellular Encoding!

Outline

- Introduction to Genetic Algorithms
- Introduction to Neural Networks
- Neuroevolution
- NEAT

NEAT

- NeuroEvolution of Augmenting Topology solve effectively the TWEANN issues thanks to:
  - Historical Markings
  - Speciation
  - Complexification

Genetic Encoding in NEAT

Genome (Genotype)

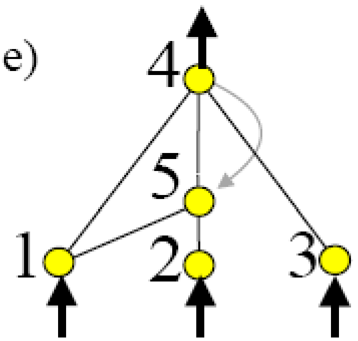
Node Genes

Node 1 Sensor	Node 2 Sensor	Node 3 Sensor	Node 4 Output	Node 5 Hidden

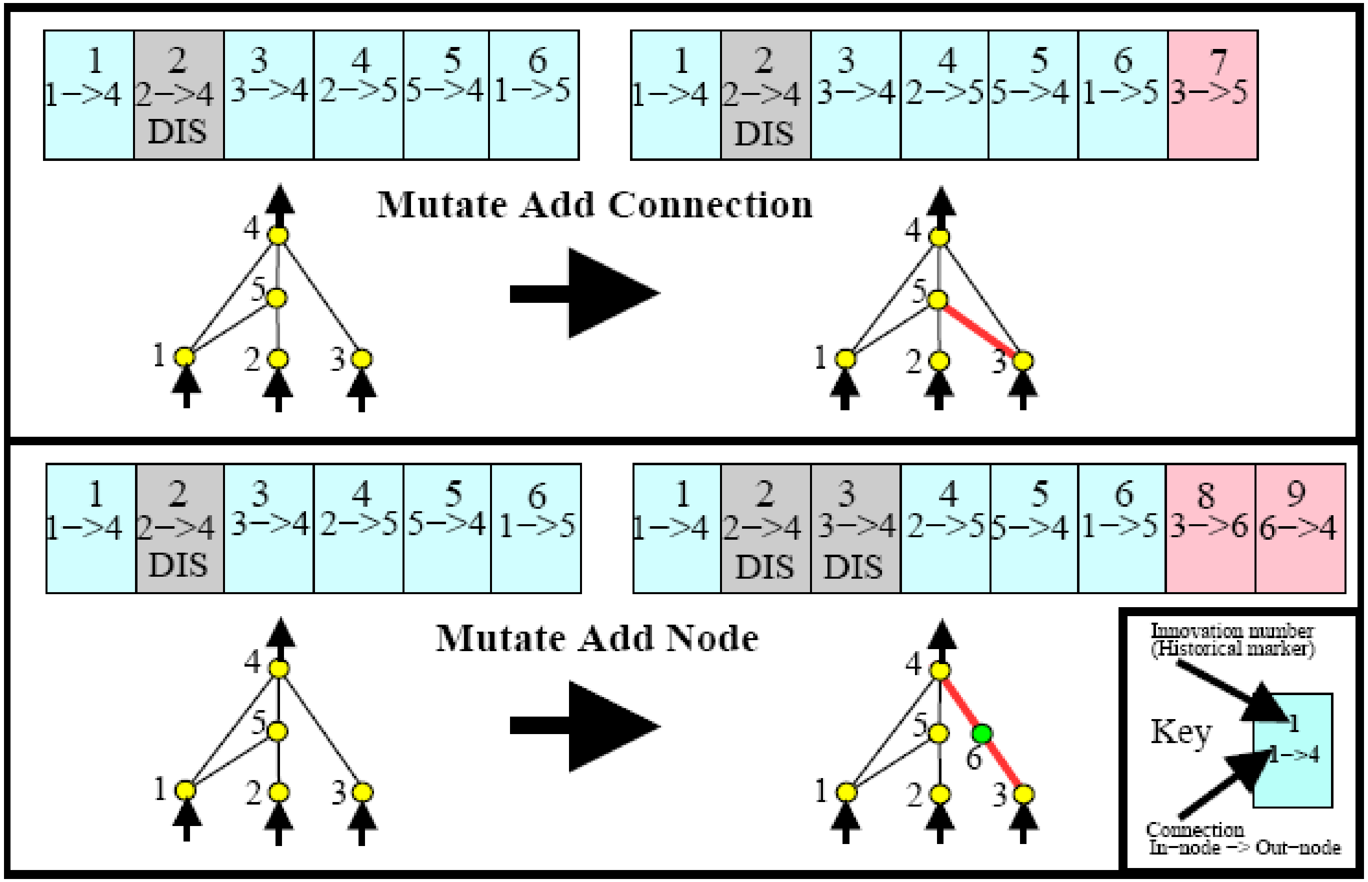
Connect. Genes

In 1	In 2	In 3	In 2	In 5	In 1	In 4
Out 4	Out 4	Out 4	Out 5	Out 4	Out 5	Out 5
Weight 0.7	Weight-0.5	Weight 0.5	Weight 0.2	Weight 0.4	Weight 0.6	Weight 0.6
Enabled	**DISABLED**	Enabled	Enabled	Enabled	Enabled	Enabled
Innov 1	Innov 2	Innov 3	Innov 4	Innov 5	Innov 6	Innov 11

Network (Phenotype)



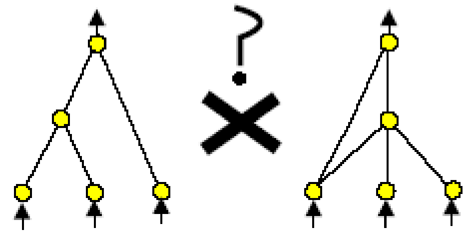
Topological Innovation



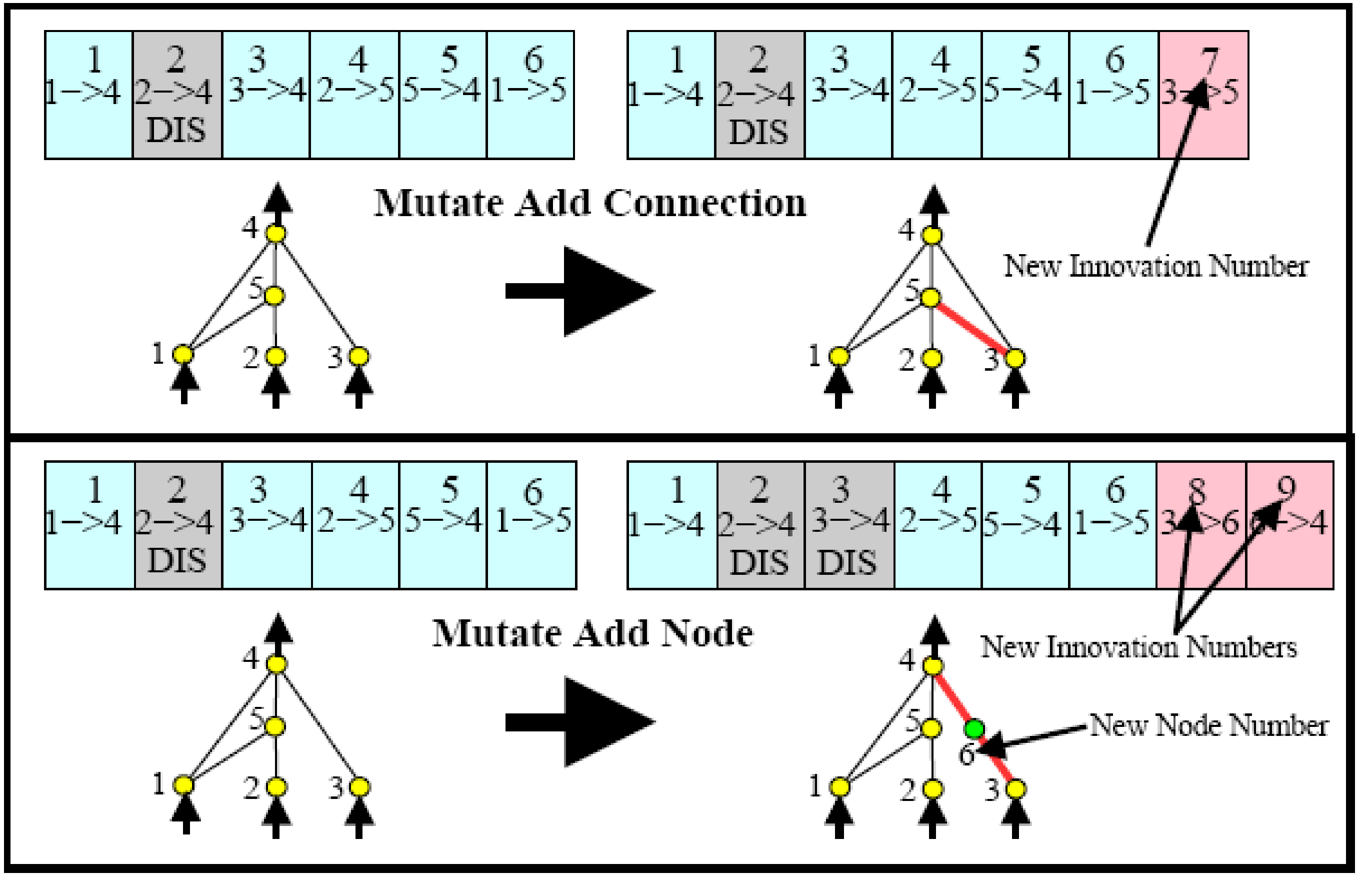
Link Weights Mutation

- A random number is added or subtracted from the current weight/parameter
- The number can be chosen from uniform, Gaussian (normal) or other distributions

Topology Matching Problem

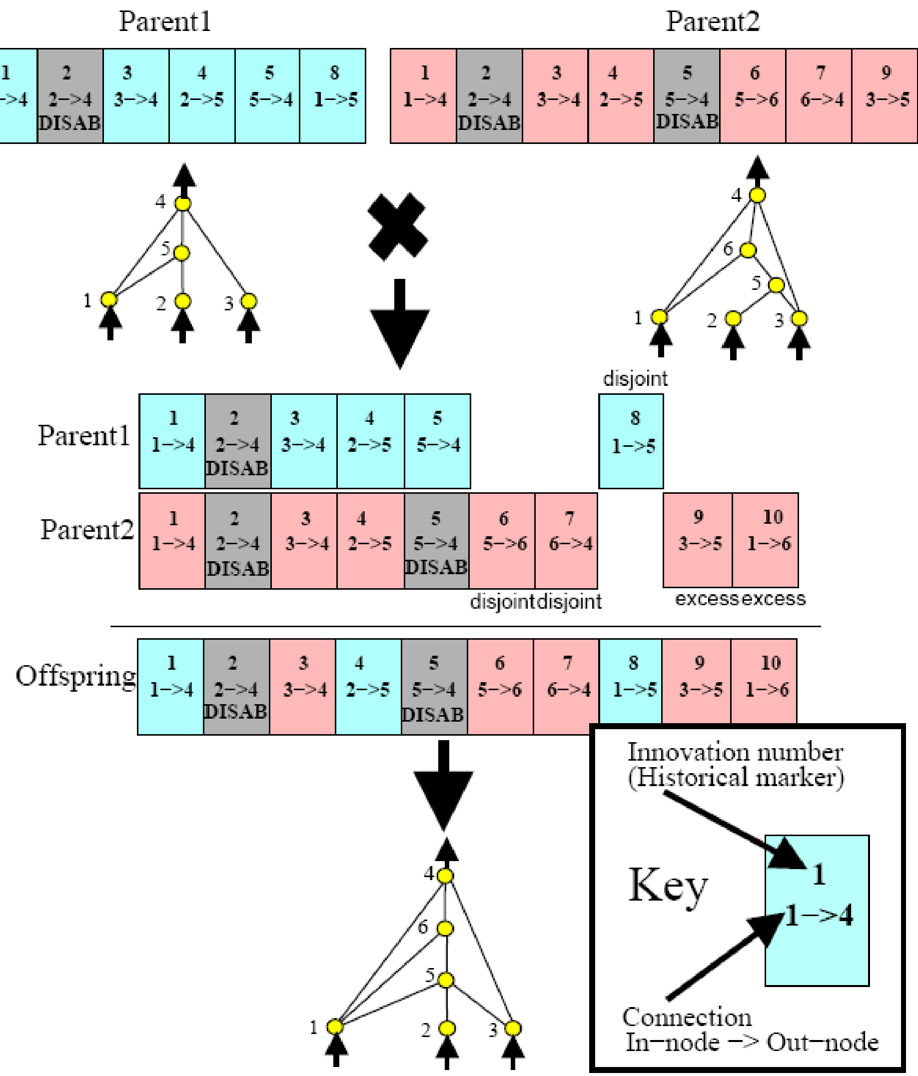


- When mating networks they may have gene with the same functionality but in different positions
- In nature **homologous** genes are aligned and matched (**synapsis**)
- How can we solve this problem ?



- Two genes with the **same history** are expected to be **homologous**

Artificial Synapsis in NEAT

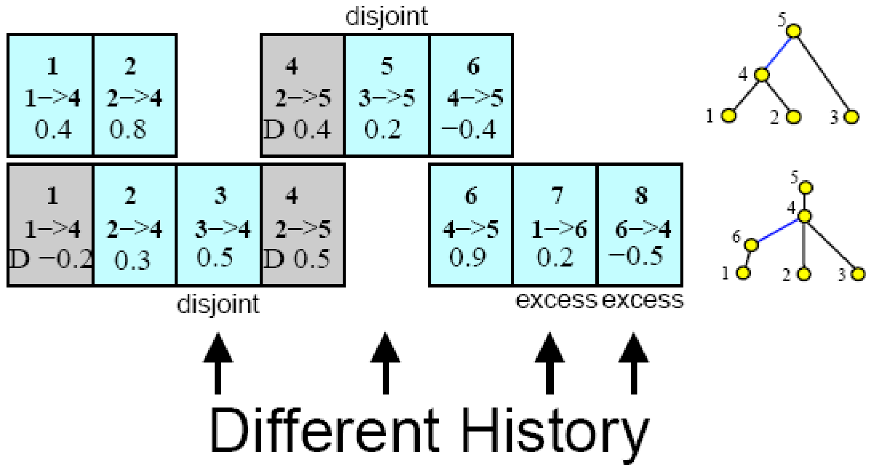


Speciation in NEAT

- At the beginning topological innovation may not be an advantage
- In order to let survive useful innovations, they have to be protected
- NEAT solve this problem with **speciation**:
  - *similar* networks are clustered
  - competition and mating is restricted to networks with the same space
  - *fitness sharing* prevent from a single species taking over the whole population

Network similarity in NEAT

- Based on historical markings
- A **compatibility distance** is computed on the basis of the excess (E), disjoint (D) and the average weight distance W of matching



$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \bar{W}$$

Fitness Sharing in NEAT

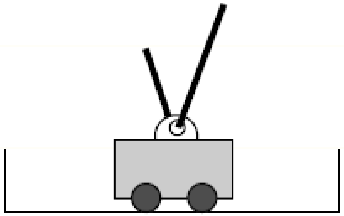
- The fitness of each network is normalized with respect to the number of networks of the same species
- At each species is assigned a number of offspring in the next generation proportional to the fitness sum of the species members.
- Offspring are generated for each species by mating the fittest r% members of the species
- Speciation gives to promising topology the chance of being optimized without the need of competing with all the other networks
- Fitness sharing prevents from a single species taking over the whole population (i.e. it keeps diversity)

Complexification

- In NEAT population is initialized with networks having the identical minimal topology (all inputs directly connected to outputs)
- More complex topology are introduced with mutation and survives only if useful
- Complexification has two advantages:
  - make trivial the initialization of the population
  - keeps limited the dimensionality of search space **through the whole** evolutionary process

Validation of NEAT

- NEAT has been tested on the double pole balancing problem (also without velocities info)



- Results shows that NEAT solves problems more effectively and faster than previous TWEANN systems
- Each of the three key components of NEAT is necessary

Applications

- Games
- Navigation
- Vehicle warning systems

Conclusions

- NEAT is an effective approach to NE thanks to
  - Historical marking
  - Speciation
  - Complexification
- Directions
  - Developmental Encoding
  - Adaptive Synapses
  - rtNEAT
  - Competitive Coevolution