Neuroevolution and Other Techniques for Generating Realistic Behavior

TAGD Presentation

November 14, 2012

Yoonsuck Choe, Ph.D. [a]

Brain Networks Lab & Neural Intelligence Lab.

Texas A&M CSE

[a] Part I&II largely based on Risto Miikkulainen's tutorial at the GECCO 2005.`http://www.cs.utexas.edu/users/risto`. Part III based on Dinesh Manocha's presentation.

How to Generate Realistic Behavior, for Games?

Call of Duty ®

Heider and Simmel [2]

- Which one looks more realistic?
- Which one will show more realistic behavior?

Outline

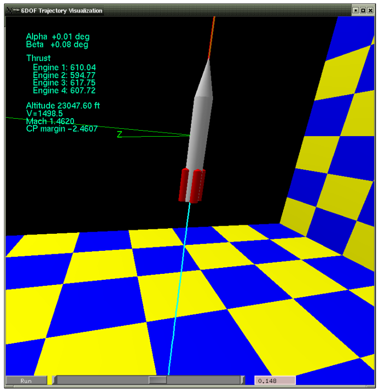
- Introduction to neuroevolution
- Evolving complex behavior through complexification and co-evolution (Stanley, Miikkulainen)
- Composite Agents (Yeh et al.) – if time permits
- Discussion

I. Intro to Neuroevolution

Neuroevolution of Complex Behavior

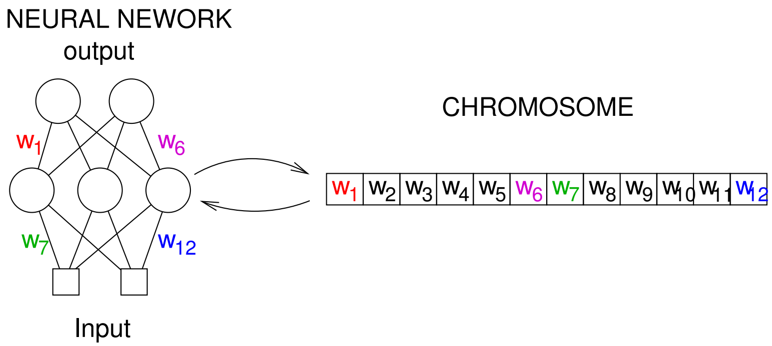
- Neuroevolution: Evolving artificial neural networks to control behavior of robots and agents.
- Main idea: Mimic the natural process of evolution that gave rise to the brain, the source of intelligence.
  - Population
  - Competition
  - Selection
  - Reproduction and mutation

Why Neuroevolution?



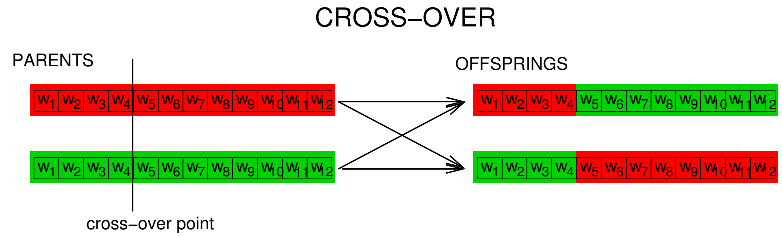
- Neural networks are effective but with limitations.
- Can solve tough, complex problems: fin-less rockets, robotic agents.

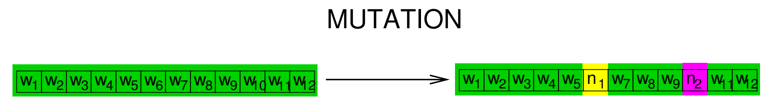
Neuroevolution Basics



- A single chromosome encodes a full neural network.
  - Inputs hooked up to sensors, and outputs to actuators.
- Each gene, a single bit (or a real number), maps to a connection weight in the neural network.

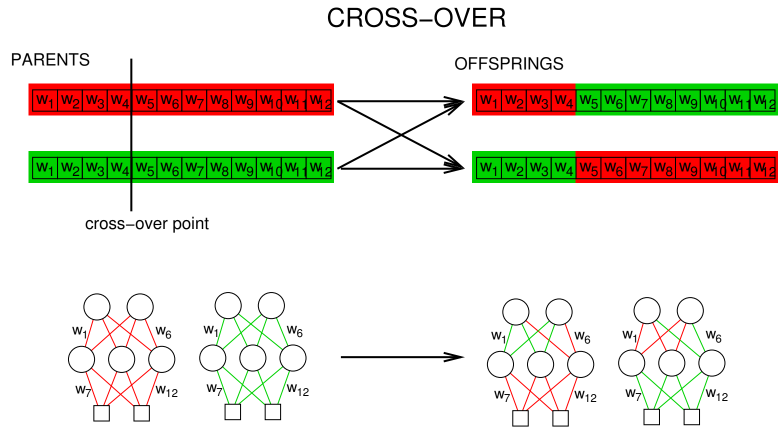
Neuroevolution Basics: Operators





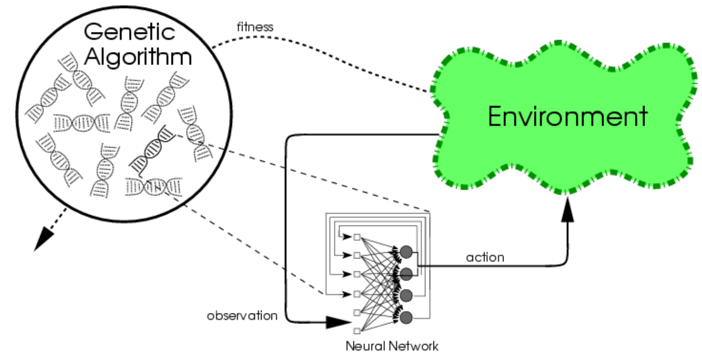
- Cross-over: Combine traits from both parents.
- Mutation: Introduce randomness (innovation).

Neuroevolution Basics: Cross-Over in Detail



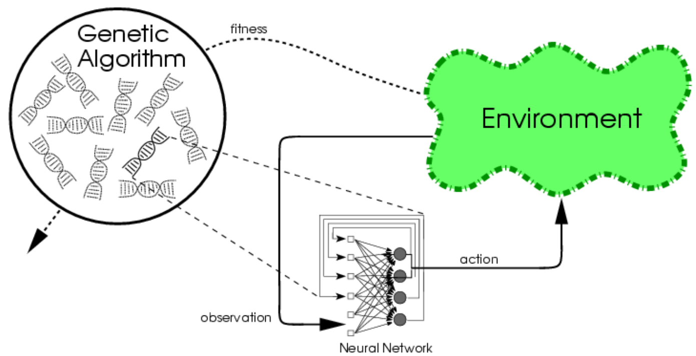
- Cross-over of two individuals produces two offsprings with a mixed heritage.

Conventional Neuroevolution (2)



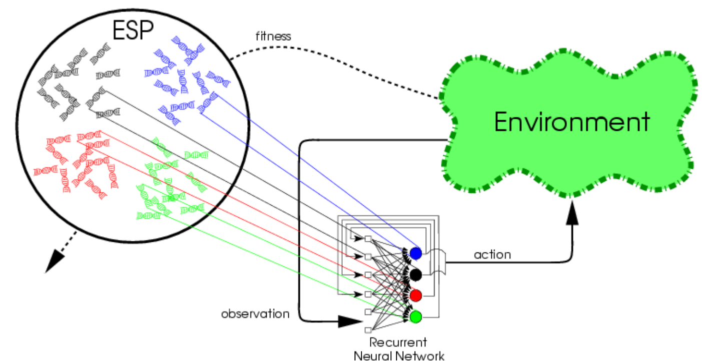
1. Fitness Evaluation: Construct NN with chromosome, put in the environment, observe outcome.
2. Selection: Choose best ones.
3. Reproduction: Mate the best ones and put back in the population.

Problems with CNE



- Evolution tends to converge to a small homogeneous population
  - Diversity is lost; progress stagnates
- Competing conventions
  - Different, incompatible encodings for the same solution
- Too many parameters to be optimized simultaneously
  - Thousands of weight values at once

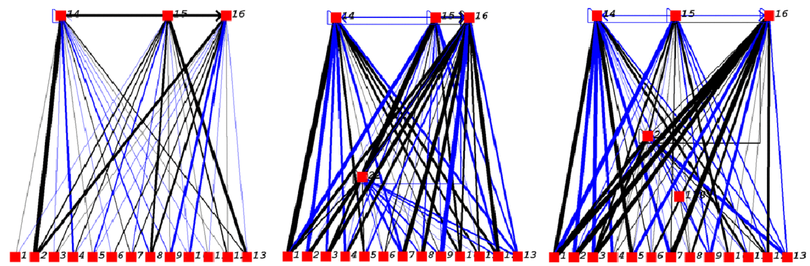
Advanced Neuroevol.: Evolving Neurons



- Evolving individual neurons: Chromosome = neuron. [1,3,4]
- Construct network with neurons, evaluate, reproduce, and repeat.
  - Network has fixed topology.
- Fitness of network determines that of participating neurons.
- Shown to improve diversity.

II. Evolving Complex Behavior: Co-Evolution & Topology Evolution[5,6]

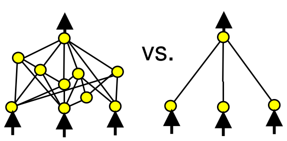
Evolving Topologies



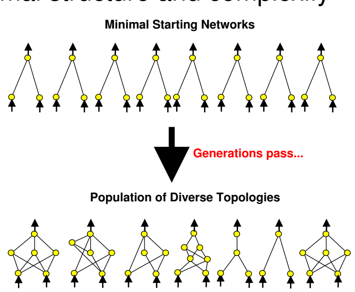
- Fixed topology has limitations.
- Idea: Evolve network topology, as well as connection weight.
- Neuroevolution of Augmenting Topologies (NEAT[5,6])
- Based on *Complexification*:
  - Network topology
  - Behavior

How Can We Complexify?

- Can optimize not just weights but also topologies



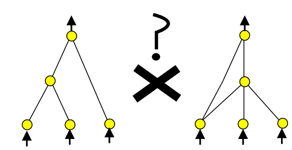
- Solution: Start with minimal structure and complexify[8]



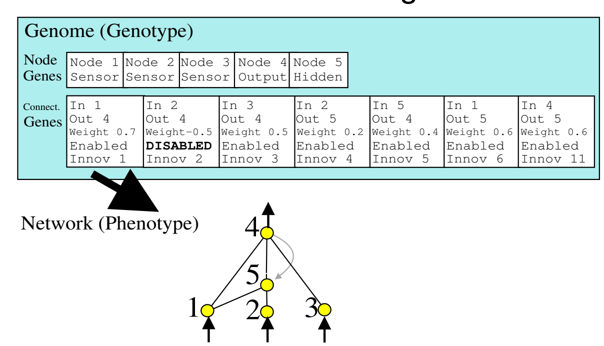
- Can search a very large space of configurations!

How Can Crossover be Implemented?

- Problem: Structures do not match

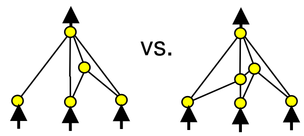


- Solution: Utilize historical markings



How can Innovation Survive?

- Problem: Innovations have initially low fitness

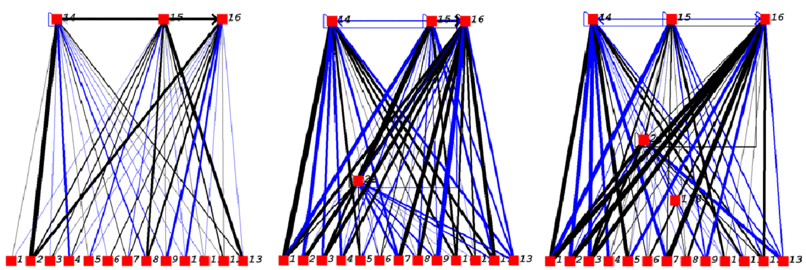


- Solution: Speciate the population
  - Innovations have time to optimize
  - Mitigates competing conventions
  - Promotes diversity

Competitive Coevolution

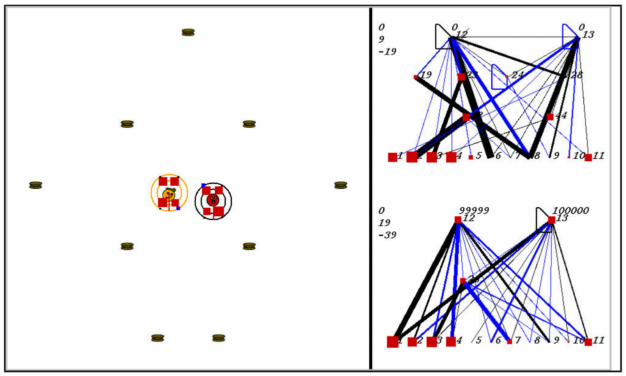
- Progress in evolution is based on competition.
- Better solutions emerge when given tougher opponents.
- Tough opponents do not exist from the beginning.
- Co-evolution solves this problem.
  - Start out with naive populations.
  - Make populations compete with each other.
  - Coevolutionary arms race (poison toxicity vs. tolerance).

Competitive Coevolution with NEAT



- Complexification elaborates on the solution
  - Adding more complexity to existing behaviors
- Can establish a coevolutionary arms race
  - Two populations continually outdo each other
  - Absolute progress, not just tricks

Coevolution Demo (by Ken Stanley)



- Two robots pitted against each other[7]
  - Food sensor, Enemy sensor, Energy difference sensor, Wall sensor
  - Eat food to incr. Energy, Moving around decr. energy.

Early Poor Strategy

- Generation 1 and 3 champs.
- Very goal-directed: eat food, attack opponent

Later Poor Strategy

- Champs from two different population in gen 40.
- No food consumption (poor strategy).
- Waste energy while idly moving (teasing?).

First Successful Strategy

- Gen 80 champ vs. Gen 95 descendant
- Switching behavior between foraging, caution, predation; Final standoff.

Old West-Style Standoff

- Gen 95 vs. gen 90 champ.
- Extended standoff

Later Dominant vs. Early Good Str.

- Gen 221 champ (later dominant strategy) vs. gen 130 champ (first good strategy).
- Caution when seeking food. Switching of strategy observed.

Highest- vs. Prior-Dominant Str.

- Gen 313 champ vs. gen 210 champ.
- Waiting until the moment is just right.
- Food nearby, enemy wasting energy, etc. all considered.

Highest Dominant vs. First Good Str.

- Gen 313 champ vs. gen 95 champ.
- Highest dominant is dominant over all past dominant.

Other Applications of NEAT

- NERO (NeuroEvolution of Robotic Operatives): Interactive neuroevolution for realtime strategy game-like environment (`http://nerogame.org`)
- Dancing, driving, generation of art, etc.
- See Ken Stanley's web page.

NERO Details

1. Approach Enemy
2. Hit Target
3. Avoid Fire
4. Approach Flag
5. Stick Together
6. Stand Guard

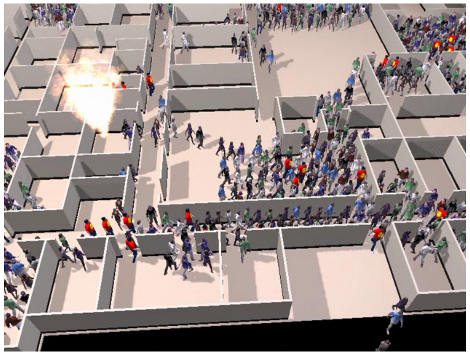
[NERO Demo]

Summary (NEAT)

- Evolving neural network topologies helps evolve complex emergent behavior.
- Co-evolution ensures continuous progress.
- Diverse applications possible.

III. Composite Agents[9]

Crowd Modeling with Composite Agents

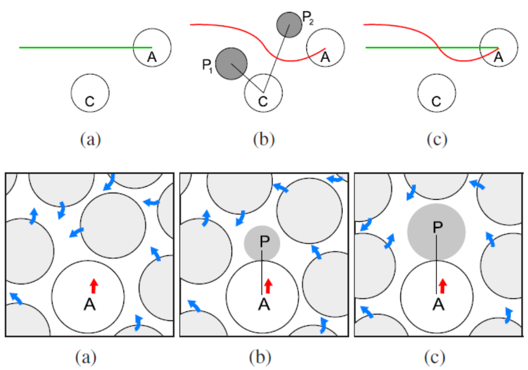


Yeh et al.[9]

A simple idea of “proxy” can:

- Help simplify task specification.
- Lead to emergent, realistic behavior.

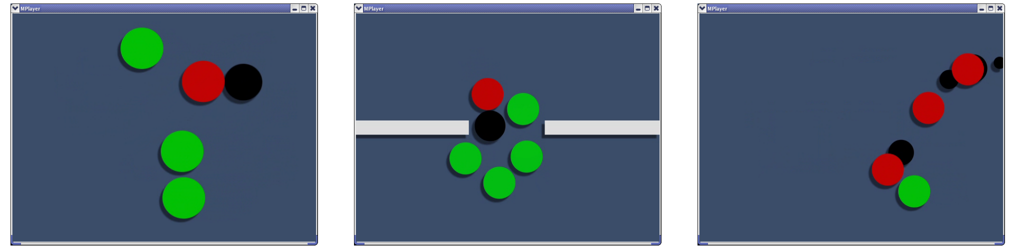
The Concept of “Proxy”



Yeh et al. [9]

- Proxies are like ghosts attached to the main agent.
- Attaching or dynamically generating “proxies” can greatly simplify behavioral modeling.

Types of Proxies



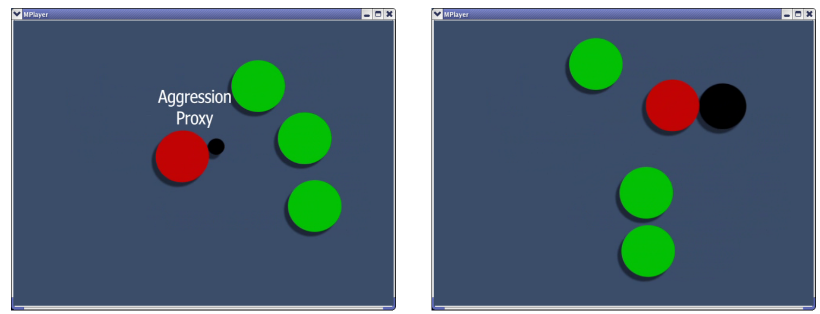
- Aggression proxy
- Priority proxy
- Trailing proxy

Use default planner with these proxies.

Proxy: Intangible Factors

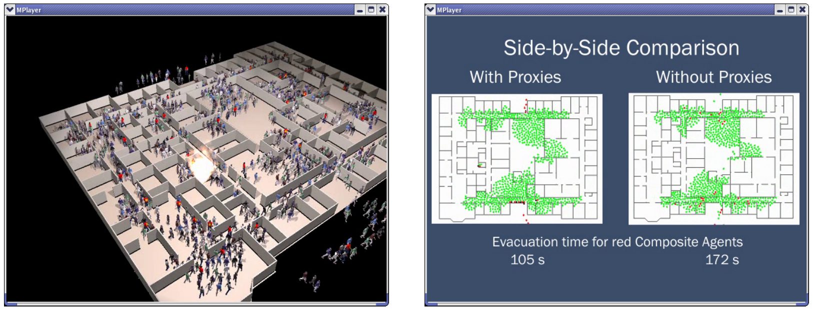
- Social and psychological factors can be translated into proxies.

Proxy: Aggression Proxy



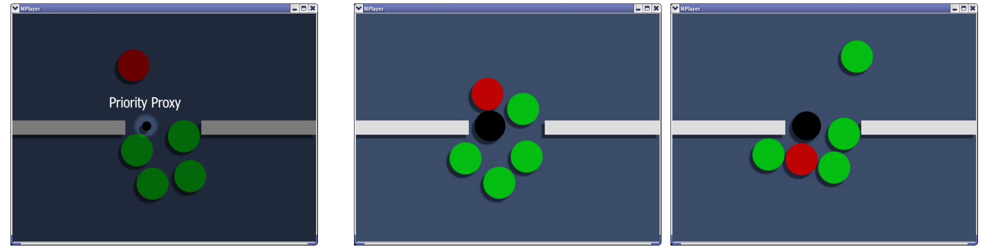
- Red: aggressor (with black proxy), Green: normal.

Proxy: Office Evacuation Example



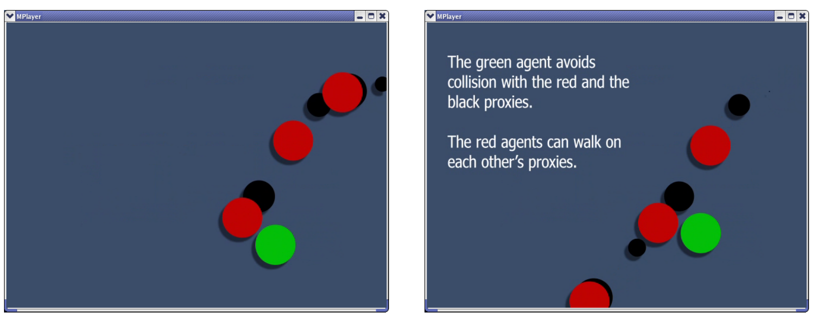
- Agents with aggression proxy faster to evacuate building.

Proxy: Priority Proxy



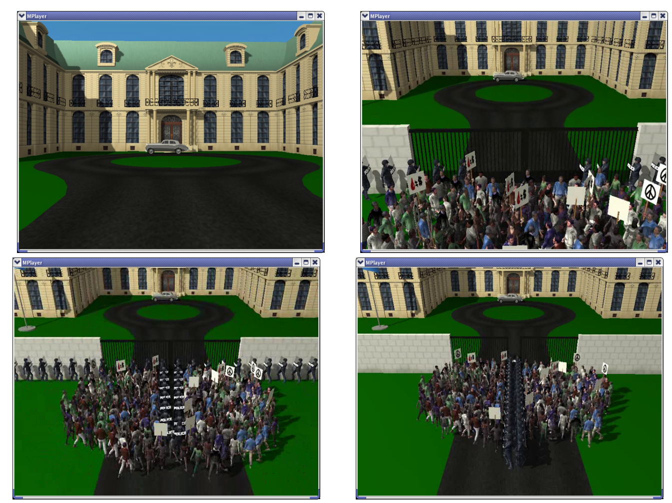
- Priority proxy implements social protocol.

Proxy: Trail Proxy



- Trail proxy enforces authority.

Proxy: Embassy Evacuation Example



- Trail proxy helps maintain police line.

DEMO

Crowd modeling with composite agents

`http://gamma.cs.unc.edu/CompAgent/CompAgent.avi`

IV. Wrap Up

Discussion and Conclusion

- Neuroevolution evolution is an effective strategy for constructing complex and realistic behavior.
- Composite agents, using various proxies, can also lead to realistic behavior.
- Analyzing the evolved networks is a challenge.

References

[1] Agogino, A., Tumer, K., and Miikkulainen, R. (2005). Efficient credit assignment through evaluation function decomposition. In *Proceedings of the Genetic and Evolutionary Computation Conference*.

[2] Heider, F., and Simmel, M. (1944). An experimental study of apparent behavior. *The American Journal of Psychology*, 57:243–259.

[3] Moriarty, D. E., and Miikkulainen, R. (1997). Forming neural networks through efficient and adaptive co-evolution. *Evolutionary Computation*, 5:373–399.

[4] Potter, M. A., and Jong, K. A. D. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8:1–29.

[5] Stanley, K. O. (2003). *Efficient Evolution of Neural Networks Through Complexification*. PhD thesis, Department of Computer Sciences, The University of Texas at Austin, Austin, TX.

[6] Stanley, K. O., and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127.

[7] Stanley, K. O., and Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100.

[8] Whiteson, S., Stone, P., Stanley, K. O., Miikkulainen, R., and Kohl, N. (2005). Automatic feature selection in neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*.

[9] Yeh, H., Curtis, S., Patil, S., van den Berg, J., Manocha, D., and Lin, M. (2008). Composite agents. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 39–47. Aire-la-Ville, Switzerland: Eurographics Association.