Advanced AI for Games: Neuroevolution

- CSCE 315
- These slides are from Risto Miikkulainen's tutorial at the GECCO 2005 conference, with slight editing.
- Slides 7 – 9 were added by Yoonsuck Choe.

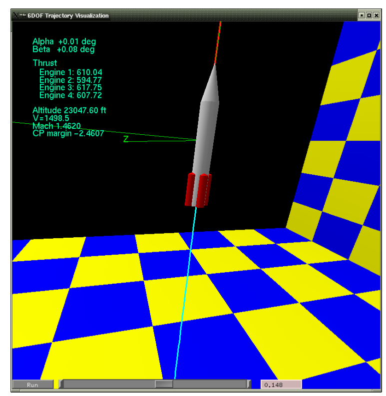
Today's Main Topic

- Neuroevolution: Evolve 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

Evolving Neural Networks

Risto Miikkulainen
Department of Computer Sciences
The University of Texas at Austin
http://www.cs.utexas.edu/users/risto

Why Neuroevolution?



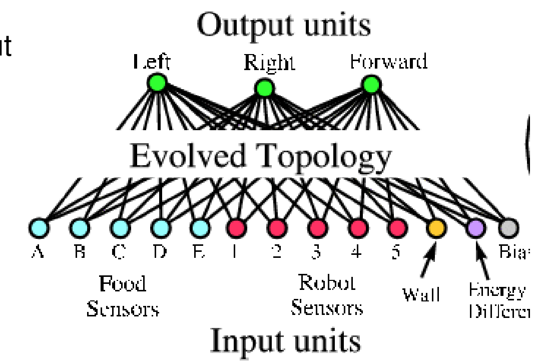
- Neural networks already successful in many domains.
- However, in certain domains, it is hard to fit the existing framework and learning algorithms.
- Hard domains: fin-less rocket control, robotic agent control, etc.

Outline

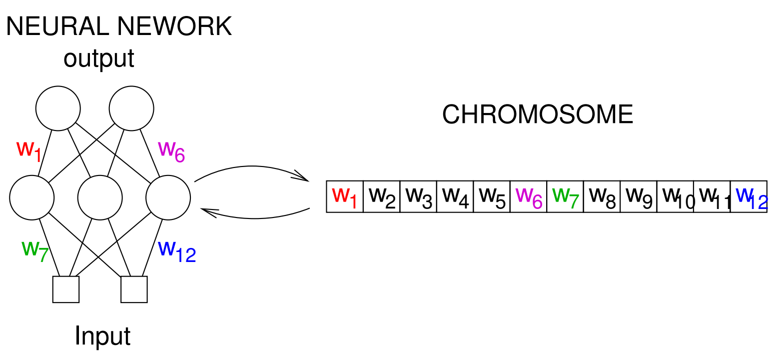
- Basic neuroevolution techniques
- Advanced techniques
  - E.g. combining learning and evolution
- Extensions to applications
- Application examples
  - Control, Robotics, Artificial Life, Games

Neuroevolution Decision Strategies

- Input variables describe the state
- Output variables describe actions
- Network between input and output
  - Hidden nodes
  - Weighted connections
- Execution:
  - Numerical activation of input
  - Nonlinear weighted sums
- Performs a nonlinear mapping
  - Memory in recurrent connections
- Connection weights and structure evolved

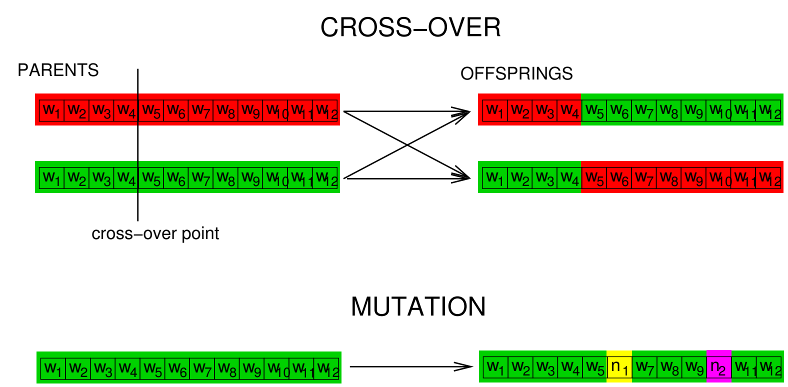


Neuroevolution Basics



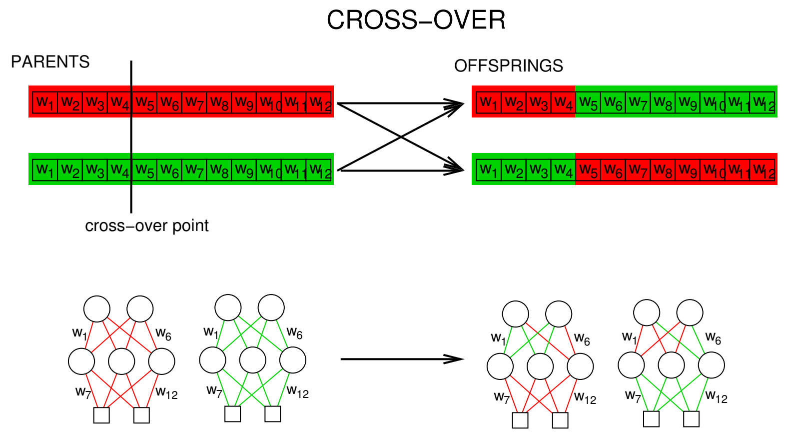
- A single chromosome encodes a full neural network.
- Each gene, a single bit (or a real number), maps to a connection weight in the neural network.

Neuroevolution Basics: Operations



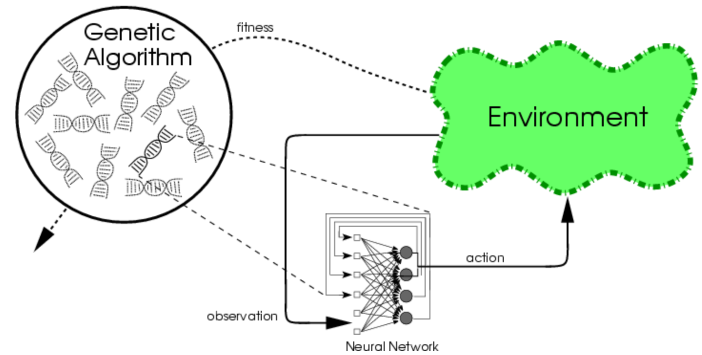
- Cross-over.
- Mutation.

Neuroevolution Basics: Cross-Over in Detail



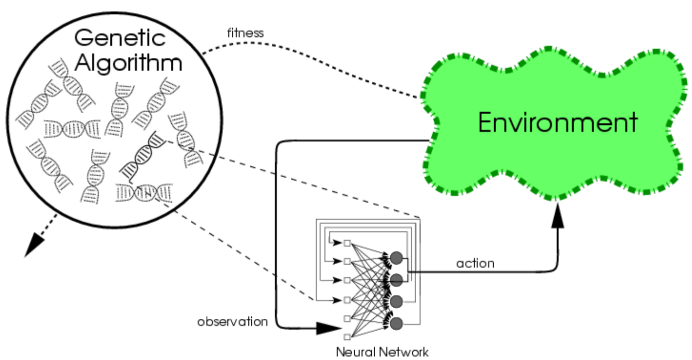
- Cross-over of two individuals produces two offsprings with a mixed heritage.

Conventional Neuroevolution (CNE)



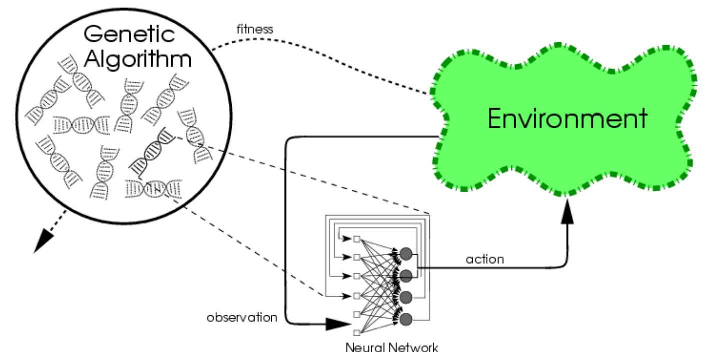
- Evolving connection weights in a population of networks [19,38,39]
- Chromosomes are strings of weights (bits or real)
  - E.g. 10010110101100101111001
  - Usually fully connected, fixed topology
  - Initially random

Conventional Neuroevolution (2)



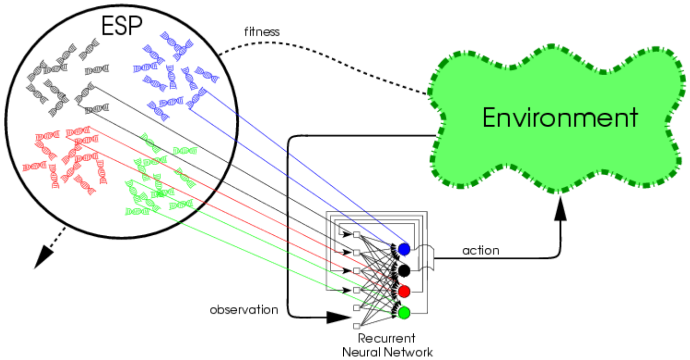
- Each NN evaluated in the task
  - Good NN reproduce through crossover, mutation
  - Bad thrown away
  - Over time, NNs evolve that solve the task
- Natural mapping between genotype and phenotype
- GA and NN are a good match!

Problems with CNE



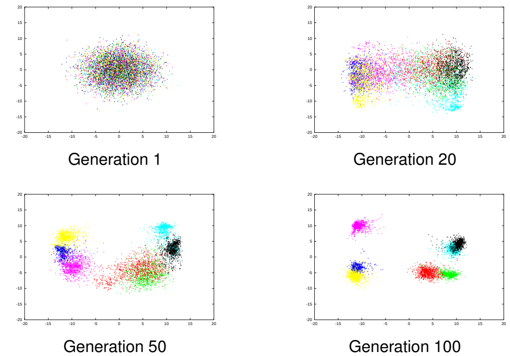
- Evolution converges the population (as usual with EAs)
  - Diversity is lost; progress stagnates
- Competing conventions
  - Different, incompatible encodings for the same solution
- Too many parameters to be optimized simultaneously
  - Thousands of weight values at once

Advanced NE 1: Evolving Neurons



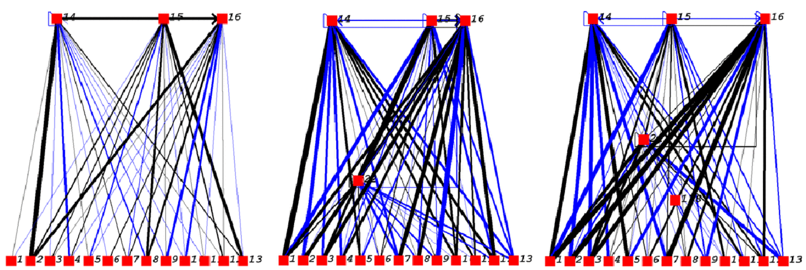
- Evolving individual neurons to cooperate in networks [1,22,24] (Agogino GECCO'05)
- E.g. Enforced Sub-Populations (ESP [?])
  - Each (hidden) neuron in a separate subpopulation
  - Fully connected; weights of each neuron evolved
  - Populations learn compatible subtasks

Advanced NE 2: Evol. Subpopulations



- Evolution encourages diversity automatically
  - Good networks require different kinds of neurons
- Evolution discourages competing conventions
  - Neurons optimized for compatible roles
- Large search space divided into subtasks
  - Optimize compatible neurons

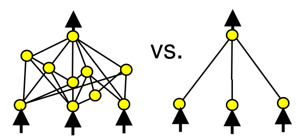
Advanced NE 3: Evolving Topologies



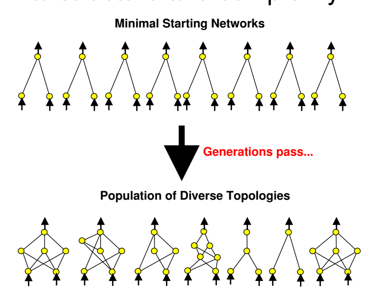
- Optimizing connection weights and network topology [11,40]
- E.g. Neuroevolution of Augmenting Topologies (NEAT [27,29])
- Based on *Complexification*
- Of networks:
  - Mutations to add nodes and connections
- Of behavior:
  - Elaborates on earlier behaviors

How Can We Complexify?

- Can optimize not just weights but also topologies



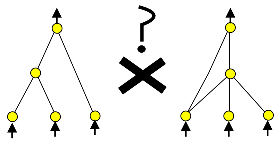
- Solution: Start with minimal structure and complexify [37]



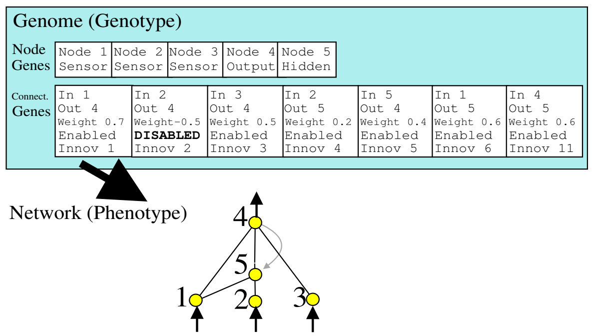
- Can search a very large space of configurations!

How Can Crossover be Implemented?

- Problem: Structures do not match

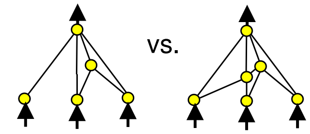


- Solution: Utilize historical markings



How can Innovation Survive?

- Problem: Innovations have initially low fitness



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

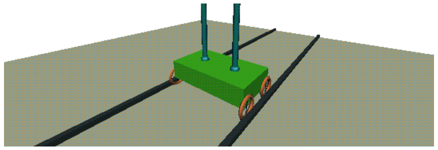
Further Neuroevolution Techniques

- Incremental evolution [13,33,39]
- Utilizing population culture [2,18]
- Evolving ensembles of NNs [16,23,36]
  (Pardoe GECCO'05)
- Evolving neural modules [25]
- Evolving transfer functions and learning rules [4,26**?**]
- Combining learning and evolution

Neuroevolution Applications

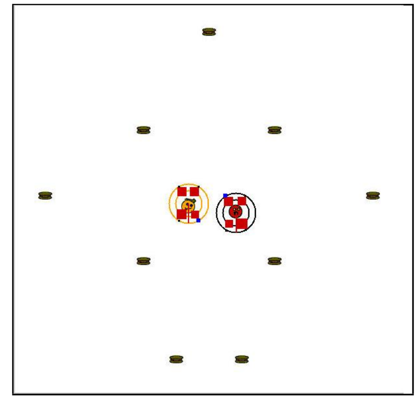
- Evolving composite decision makers [36]
- Evolving teams of agents [3,28,41]
- Utilizing coevolution [30]
- Real-time neuroevolution [28]
- Combining human knowledge with evolution [8]

Applications to Control



- Pole-balancing benchmark
  - Originates from the 1960s
  - Original 1-pole version too easy
  - Several extensions: acrobat, jointed, 2-pole, particle chasing[23]
- Good surrogate for other control tasks
  - Vehicles and other physical devices
  - Process control[34]

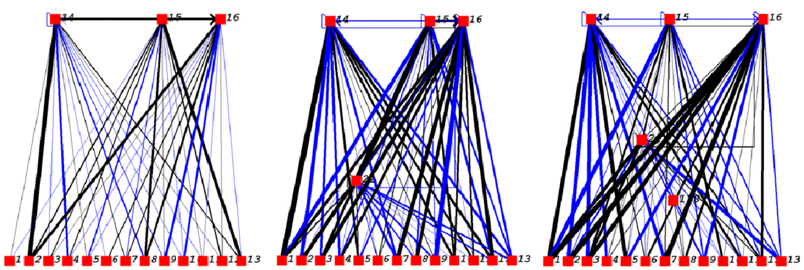
Competitive Coevolution



- Evolution requires an opponent to beat
- Such opponents are not always available

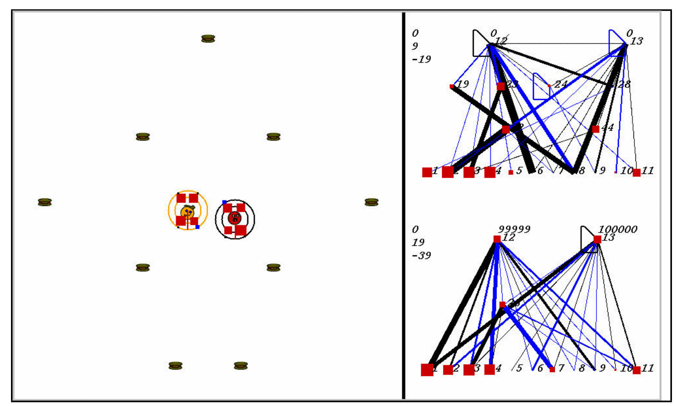
- Co-evolve two populations to outdo each other
- How to maintain an arms race?

Competitive Coevolution with NEAT



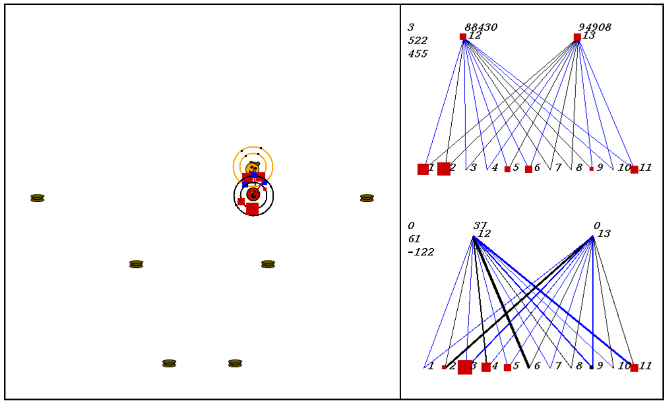
- Complexification elaborates instead of alters
  - Adding more complexity to existing behaviors
- Can establish a coevolutionary arms race
  - Two populations continually outdo each other
  - Absolute progress, not just tricks

Robot Duel Domain



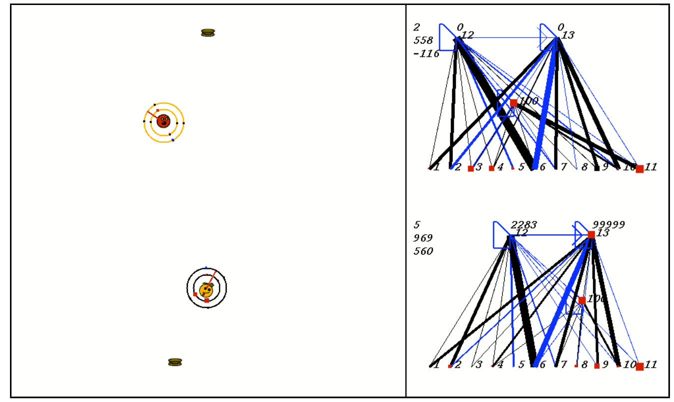
- Two Khepera-like robots forage, pursue, evade[30]
  - Collect food to gain energy
  - Win by crashing to a weaker robot

Early Strategies



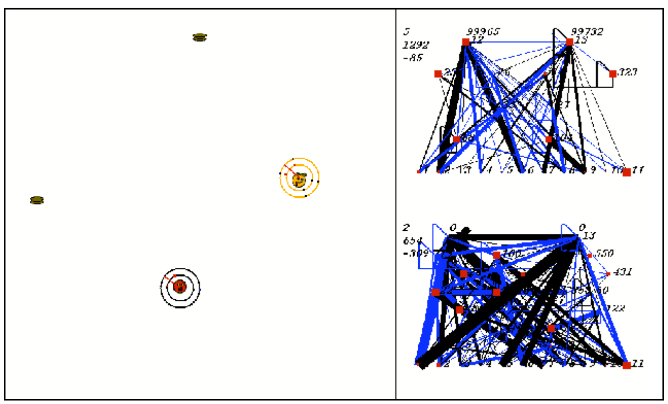
- Crash when higher energy
- Collect food by accident
- DEMO

Mature Strategies



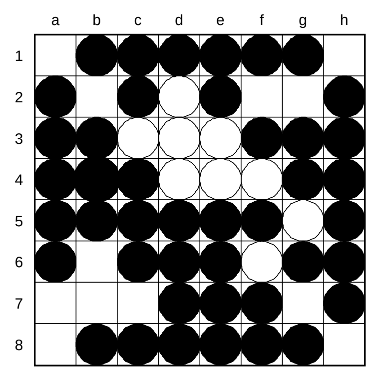
- Collect food to gain energy
- Avoid moving to lose energy
- Standoff: Difficult to predict outcome
- DEMO

Sophisticated Strategy



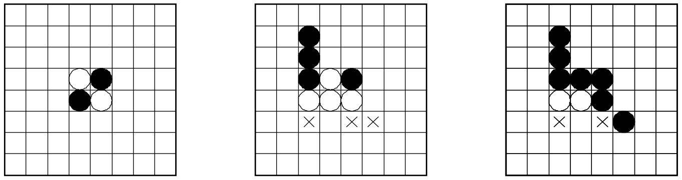
- “Fake” a move up, force away from last piece
- Win by making a dash to last piece
- Complexification → arms race
- DEMO

Applications to Games



- Good research platform
  - Controlled domains, clear performance, safe
  - Economically important; training games possible
- Board games: beyond limits of search
  - Evaluation functions in checkers, chess[5,9,10]
  - Filtering information in go[28], othello[20,31]

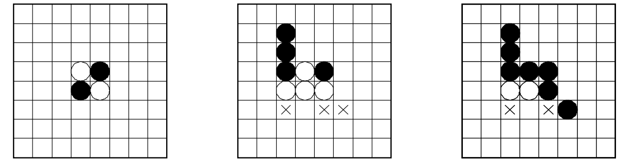
Discovering Novel Strategies in Othello



(a) (b) (c)

- Players take turns placing pieces
- Each move must flank opponent's piece
- Surrounded pieces are flipped
- Player with most pieces wins

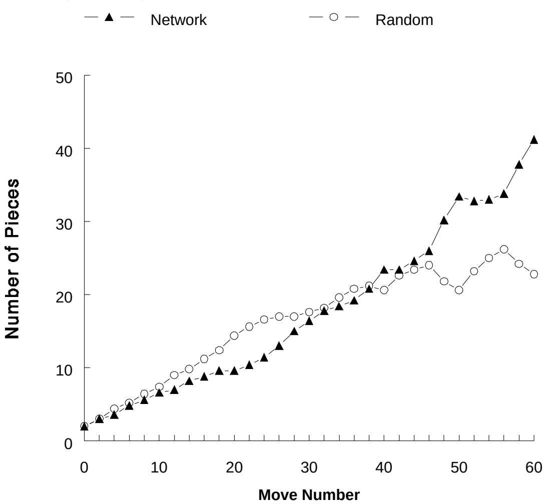
Strategies in Othello



(a) (b) (c)

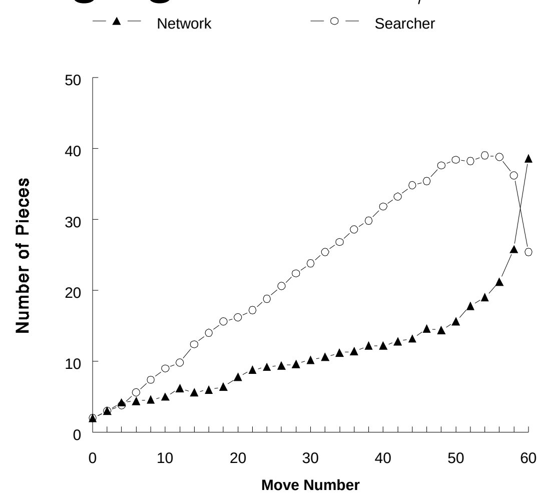
- Positional
  - Number of pieces and their positions
  - Typical novice strategy
- Mobility
  - Number of available moves: force a bad move
  - Much more powerful, but counterintuitive
  - Discovered in 1970's in Japan

Evolving Against a Random Player



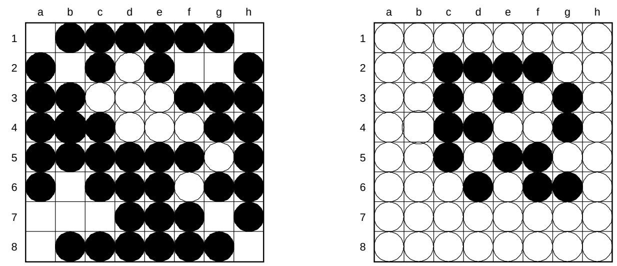
- Network sees the board, suggests moves by ranking[21]
- Networks maximize piece counts throughout the game
- A positional strategy emerges
- Achieved 97% winning percentage

Evolving Against an α-β Program



- Iago's positional strategy destroyed networks at first
- Evolution turned low piece count into an advantage
- Mobility strategy emerged!
- Achieved 70% winning percentage

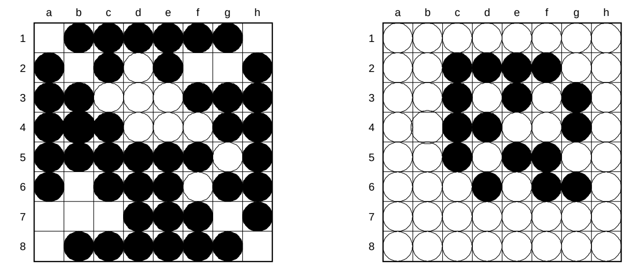
Example game



(a) (b)

- Black's positions strong, but mobility weak
- White (the network) moves to f2
- Black's available moves b2, g2, and g7 each will surrender a corner
- The network wins by forcing a bad move

Discovering Novel Strategies



(a) (b)

- Neuroevolution discovered a strategy novel to us
- "Evolution works by tinkering"
  - So does neuroevolution
  - Initial disadvantage turns into novel advantage

Future Challenge: Utilizing Knowledge

- Given a problem, NE discovers a solution by exploring
  - Sometimes you already know (roughly) what works
  - Sometimes random initial behavior is not acceptable
- How can domain knowledge be utilized?
  - By incorporating rules (Yong GECCO'05)
  - By learning from examples

Numerous Other Applications

- Creating art, music[6]
- Theorem proving[7]
- Time-series prediction[17]
- Computer system optimization[12]
- Manufacturing optimization[14]
- Process control optimization[34,35]
- Etc.

Conclusion

- Neuroevolution, mimicing the natural process of evolution, is an effective strategy for constructing complex and useful behavior.
- Neuroevolution often performs well for reinforcement learning tasks.
- Analyzing the resulting network is a challenge.

References

[1] Adrian Agogino, Kagan Tumer, and Risto Miikkulainen. Efficient credit assignment through evaluation function decomposition. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2005.

[2] Richard K. Belew. Evolution, learning and culture: Computational metaphors for adaptive algorithms. *Complex Systems*, 4:11–49, 1990.

[3] Bobby D. Bryant and Risto Miikkulainen. Neuroevolution for adaptive teams. In *Proceedings of the 2003 Congress on Evolutionary Computation*, 2003.

[4] David J. Chalmers. The evolution of learning: An experiment in genetic connectionism. In Touretzky et al. [32], pages 81–90.

[5] Kumar Chellapilla and David B. Fogel. Evolution, neural networks, games, and intelligence. *Proceedings of the IEEE*, 87:1471–1496, 1999.

[6] Chun-Chi Chen and Risto Miikkulainen. Creating melodies with evolving recurrent neural networks. In *Proceedings of the INNS-IEEE International Joint Conference on Neural Networks*, pages 2241–2246, Piscataway, NJ, 2001. IEEE.

[7] Nirav S. Desai and Risto Miikkulainen. Neuro-evolution and natural deduction. In *Proceedings of The First IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, pages 64–69, Piscataway, NJ, 2000. IEEE.

[8] James Fan, Raymond Lau, and Risto Miikkulainen. Utilizing domain knowledge in neuroevolution. In *Machine Learning: Proceedings of the 20th Annual Conference*, 2003.

[9] David B. Fogel. *Blondie24: Playing at the Edge of AI*. Kaufmann, San Francisco, 2001.

[10] David B. Fogel, Timothy J. Hays, Sarah L. Hahn, and James Quon. Further evolution of a self-learning chess program. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, Piscataway, NJ, 2005. IEEE.

[11] Brad Fullmer and Risto Miikkulainen. Using marker-based genetic encoding of neural networks to evolve finite-state behaviour. In Francisco J. Varela and Paul Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 255–262. MIT Press, Cambridge, MA, 1992.

[12] Faustino Gomez, Doug Burger, and Risto Miikkulainen. A neuroevolution method for dynamic resource allocation on a chip multiprocessor. In *Proceedings of the INNS-IEEE International Joint Conference on Neural Networks*, pages 2355–2361, Piscataway, NJ, 2001. IEEE.

[13] Faustino Gomez and Risto Miikkulainen. Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342, 1997.

[14] Brian Greer, Henri Hakonen, Risto Lahdelma, and Risto Miikkulainen. Numerical optimization with neuroevolution. In *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 361–401, Piscataway, NJ, 2002. IEEE.

[15] Christian Igel. Neuroevolution for reinforcement learning using evolution strategies. In *Proceedings of the 2003 Congress on Evolutionary Computation*, pages 2588–2595, 2003.

[16] Yong Liu, Xin Yao, and Tetsuya Higuchi. Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, 4:380–387, 2000.

[17] J. R. McDonnell and D. Waagen. Evolving recurrent perceptrons for time-series modeling. *IEEE Transactions on Evolutionary Computation*, 5:24–38, 1994.

[18] Paul McQuesten. *Cultural Enhancement of Neuroevolution*. PhD thesis, Department of Computer Sciences, The University of Texas at Austin, Austin, TX, 2002. Technical Report AI-02-295.

[19] David J. Montana and Lawrence Davis. Training feedforward neural networks using genetic algorithms. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 762–767. San Francisco: Kaufmann, 1989.

[20] David E. Moriarty. *Symbiotic Evolution of Neural Networks in Sequential Decision Tasks*. PhD thesis, Department of Computer Sciences, The University of Texas at Austin, 1997. Technical Report UT-AI97-257.

[21] David E. Moriarty and Risto Miikkulainen. Discovering complex Othello strategies through evolutionary neural networks. *Connection Science*, 7(3):195–209, 1995.

[22] David E. Moriarty and Risto Miikkulainen. Forming neural networks through efficient and adaptive co-evolution. *Evolutionary Computation*, 5:373–399, 1997.

[23] David Pardoe, Michael Ryoo, and Risto Miikkulainen. Evolving neural network ensembles for control problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2005.

[24] Mitchell A. Potter and Kenneth A. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8:1–29, 2000.

[25] Joseph Reisinger, Kenneth O. Stanley, and Risto Miikkulainen. Evolving reusable neural modules. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2004.

[26] Thomas Philip Runarsson and Magnus Thor Jonsson. Evolution and design of distributed learning rules. In *Proceedings of The First IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, pages 59–63, Piscataway, NJ, 2000. IEEE.

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

[28] Kenneth O. Stanley, Bobby Bryant, and Risto Miikkulainen. Real-time neuroevolution in the NERO video game. *IEEE Transactions on Evolutionary Computation*, 9:653–668, 2005.

[29] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127, 2002.

[30] Kenneth O. Stanley and Risto Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100, 2004.

[31] Kenneth O. Stanley and Risto Miikkulainen. Evolving a roving eye for go. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2004.

[32] David S. Touretzky, Jeffrey L. Elman, Terrence J. Sejnowski, and Geoffrey E. Hinton, editors. *Proceedings of the 1990 Connectionist Models Summer School*. San Francisco: Kaufmann, 1990.

[33] Joseba Urzelai, Dario Floreano, Marco Dorigo, and Marco Colombetti. Incremental robot shaping. *Connection Science*, 10:341–360, 1998.

[34] Alex v. E. Conradie, Risto Miikkulainen, and C. Aldrich. Adaptive control utilising neural swarming. In *Proceedings of the Genetic and Evolutionary Computation Conference*. San Francisco: Kaufmann, 2002.

[35] Alex v. E. Conradie, Risto Miikkulainen, and C. Aldrich. Intelligent process control utilizing symbiotic memetic neuro-evolution. In *Proceedings of the 2002 Congress on Evolutionary Computation*, 2002.

[36] Shimon Whiteson, Nate Kohl, Risto Miikkulainen, and Peter Stone. Evolving keepaway soccer players through task decomposition. *Machine Learning*, 59:5–30, 2005.

[37] Shimon Whiteson, Peter Stone, Kenneth O. Stanley, Risto Miikkulainen, and Nate Kohl. Automatic feature selection in neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2005.

[38] Darrell Whitley, Stephen Dominic, Rajarshi Das, and Charles W. Anderson. Genetic reinforcement learning for neurocontrol problems. *Machine Learning*, 13:259–284, 1993.

[39] Alexis P. Wieland. Evolving controls for unstable systems. In Touretzky et al. [32], pages 91–102.

[40] Xin Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.

[41] Chern Han Yong and Risto Miikkulainen. Cooperative coevolution of multi-agent systems. Technical Report AI01-287, Department of Computer Sciences, The University of Texas at Austin, 2001.