Neuroevolution: Evolving Neural Networks

Kenneth O. Stanley
Dept. of Electrical Engineering and Computer Science
University of Central Florida
kstanley@eecs.ucf.edu
http://www.cs.ucf.edu/~kstanley

E P L E X
Evolutionary Complexity Research Group at UCF

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

1

Instructor/Presenter

- Ken Stanley's connections to neuroevolution:
  - Co-inventor of NEAT (with Risto Miikkulainen)
  - Co-inventor of HyperNEAT (with David D'Ambrosio and Jason Gauci)
  - Co-inventor of novelty search (with Joel Lehman)
  - Co-founder of GECCO GDS Track in 2007 and Co-chair of track from 2007-2009
  - Over 50 publications in neuroevolution

2

Course Agenda

- Neuroevolution basics
- Fixed-topology evolution
- Evolving topologies and weights
- Indirect encoding of neural networks
- Advanced topics
- Demonstrations
- Future prospects and conclusions

3

Objectives of the Tutorial

- At the end, you will know:
  - What neuroevolution is about
  - Motivation for neuroevolution
  - Historical background
  - Popular approaches
  - Recent approaches
  - Current research directions
  - Major challenges ahead

4

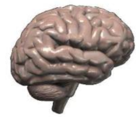
Quiz

What is the most complex artifact in the known universe?

5

Quiz

What is the most complex artifact in the known universe?



6

Quiz

What is the most complex artifact in the known universe?

100 *trillion* connections

7

Quiz

What is the most complex artifact in the known universe?

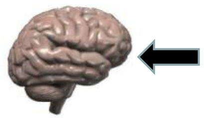
100 *trillion* connections

How did it get here?

8

Quiz

What is the most complex artifact in the known universe?



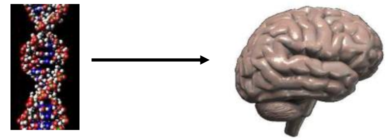
100 *trillion* connections

How did it get here?

Evolution

9

Main Idea: Combine EC and Neural Networks



- "Evolving brains":
  Neural networks compete and evolve
- Idea dates back to the 1980s [40,51,72,73]
- Natural path to AI: Only way that intelligence ever really was created
- Leads to many research challenges

10

Why Neuroevolution (NE)?

- If a brain can do it, a neural network can be evolved to do it too (many applications)
  - Supervised classification [76]
  - Autonomous control
    - Robots [5,15,31,42,45,54,56,69]
    - Vehicles [34]
    - Video game characters [27,59]
  - Factory optimization [23]
  - Game playing [19,41,64]
    - Go, Tic-tac-toe, Othello, Checkers
  - Warning systems [34]
  - Visual recognition, roving eyes [33,64]
  - Creative applications like art and music [28,29,52]

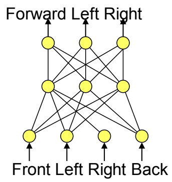
11

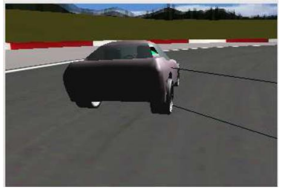
Why Neuroevolution (NE)? (2)

- Neural networks successful in many domains where no good theory exists
  - Control, pattern recognition, prediction, decision making
- Early researchers saw NE as a competitor for backpropagation (supervised learning)
  - But much more interesting when correct outputs are *not* known (fewer algorithms)

12

Sequential Decision Tasks

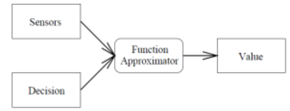




- POMDP: Sequence of decisions creates a sequence of states
- No targets: Performance evaluated after several decisions (sparse reinforcement)
- Many important real-world domains:
  - Robot/vehicle/traffic control
  - Computer/manufacturing/process optimization
  - Game playing

13

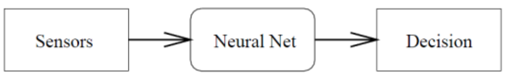
Temporal Difference Reinforcement Learning



- Q-learning, SARSA, others (state-action-space search) [65]
  - Generate targets through prediction errors
  - Learn when successive predictions differ
- Predictions represented as a value function
  - Values of alternatives at each state
- Difficult with large/continuous state and action spaces
- Difficult with hidden states (partial observability)
- NE is different…

14

Neuroevolution (NE) Reinforcement Learning



- Direct nonlinear mapping from sensors to actions (policy search)
- Large/continuous states and actions easy
  - Generalization in neural networks
- Hidden states disambiguated through memory
  - Recurrence in neural networks [66]

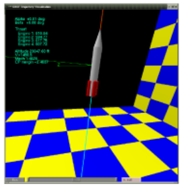
15

More than Just RL

- Big questions: How did evolution discover an artifact of such incredible complexity?
  - Can EC do something similar?
  - Why hasn't it done so yet?
- Progress means shedding light on deep issues in both EC and biology

16

Diverse Applications



Rocket Control [22]

Video Game NPC Control [59]

Evolving Pictures [52,53]

Evolving Music [28,29]

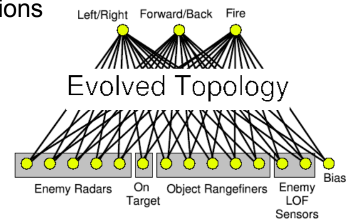
Real-world Robot Control [34]

Video Game Content Generation [27]

17

Typical ANN for NE Control

- Input variables describe the state
- Output variables describe actions
- Networks between input and output:
  - Nonlinear hidden nodes
  - Weighted connections

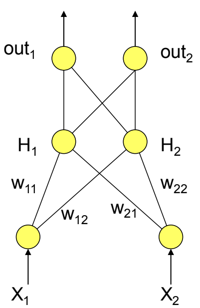


- Execution:
  - Numerical activation of input
  - Performs nonlinear mapping
  - Memory in recurrent connections

18

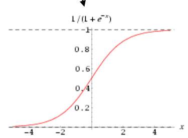
Inside the ANN

- *Simple Network Activation*



Neuron *j* activation:

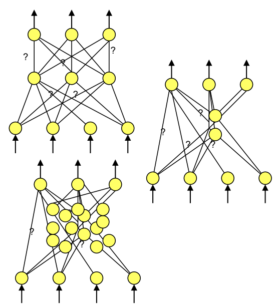
$$H_j = \sigma\left(\sum_{i=1}^{n} x_i w_{ij}\right)$$

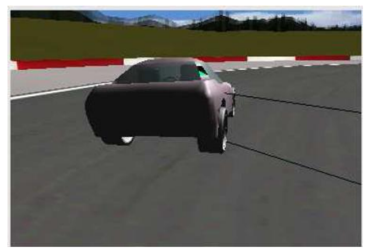


19

The Problem of Learning

- What is the topology that works?
- What are the weights that work?

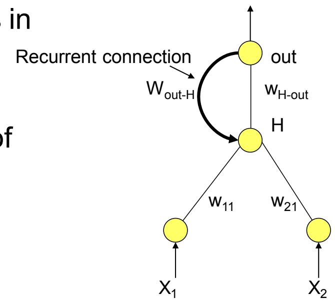




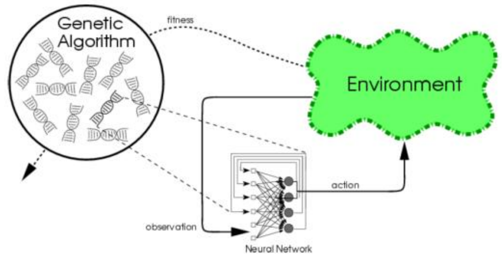
20

Recurrent Connections

- Recurrent connections are backward connections in the network
- They allow *feedback*
- Recurrence is a type of memory



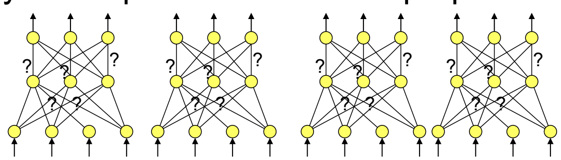
Conventional Neuroevolution [40,51,72,73]



- Genome encodes ANN
- ANN evaluated in environment for fitness
- Fitness drives selection and reproduction

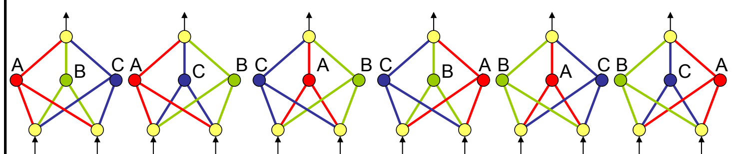
Earliest NE Methods Only evolved Weights

- Genome is a direct encoding
- Genes represent a vector of weights
- Could be a bit string or real valued
- NE optimizes the weights for the task
- Maybe a replacement for backprop



The Competing Conventions Problem [48,51]

- Also called *permutation problem*
- Many permutations of same vector represent exactly the same functionality
- Then how can crossover work?



3!=6 permutations of the same network!

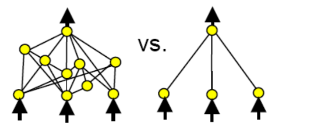
Competing Conventions Destroys Crossover

- n! permutations of an n-hidden-node 1-layer net
- [A,B,C] X [C,B,A] can be [C,B,C]
- 144 total possible crossovers of size 3
- 72 are trivial (offspring is a duplicate)
- 48 of the remaining 72 are defective
- 66.6% of nontrivial mating is defective!
- Consider also differing conventions:
  – [A,B,C]X[D,B,E]
  – Loss of coherence in GA is severe

25

Problem Dimensionality

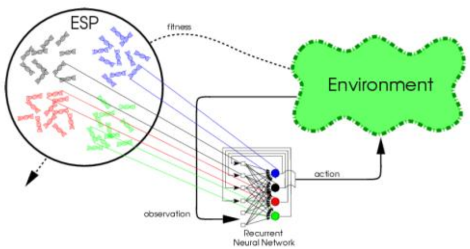
- Each connection (weight) in the network is a *dimension* in a search space



- The space you're in matters: Optimization is not the only issue!
  – Harder to optimize more dimensions
- Topology defines the space

26

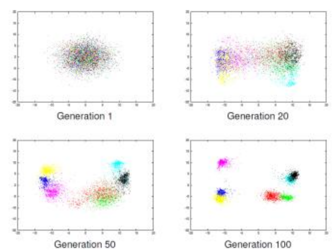
Advanced Fixed-Topology NE



- Evolving individual neurons to cooperate in networks [2,43,46]
- Example: Enforced Sub-Populations (ESP) [20]
  – Each (hidden) neuron in a separate subpopulation
  – Fully connected; weights of each neuron evolved
  – Populations learn compatible subtasks

27

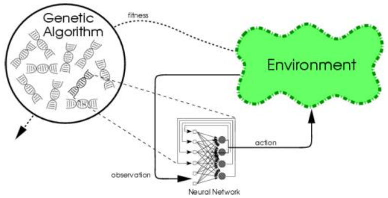
Evolving Neurons with ESP



- 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

28

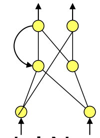
Advanced Fixed-Topology NE (2): Evolution Strategies



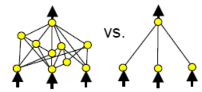
- Evolving complete networks with ES (CMA-ES) [32]
- Small populations, no crossover
- Instead, intelligent mutations
  - Adapt covariance matrix of mutation distribution
  - Take into account correlations between weights

29

TWEANNS



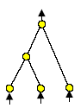
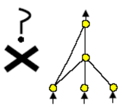
- "Topology and Weight Evolving Artificial Neural Networks" [3,14,17,61,75]
- Population contains diverse topologies
- Why leave anything to humans?
- Topology can be represented many ways
- Topology evolution can combine w/ backprop
- Remember*: Topology defines the search space*
- The more connections, the more dimensions



30

"Competing Conventions" with Arbitrary Topologies

- *Topology matching problem*
- Life is even worse with mating arbitrary topologies
- How do they match up?

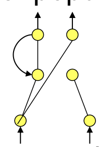



- Radcliffe (1993) : "Holy Grail in this area." [48]

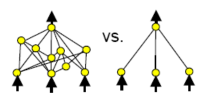
31

More TWEANN Problems

- Diverse topologies present many problems
- How should evolution begin? Randomly?
  - Defects in the initial population



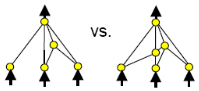
  - Searching in unnecessarily large space



32

More TWEANN Problems 2

- Innovative structures have more connections
- Innovative structure cannot compete with simpler ones



- Yet the money is on innovation in the long run
- Need some kind of protection for innovation

33

Many Early TWEANNs

- Breeder Genetic Programming [76]
  - Network is tree
  - Penalizes complexity in fitness function
- Parallel Distributed Genetic Programming (PDGP) [47]
  - Dual representation: Linear and graph
- GeNeralized Acquisition of Recurrent Links (GNARL) [4]
  - Gave up on crossover (competing conventions too problematic)
- Most began evolution with random topologies
- Often tested on supervised learning problems
  - E.g. parity & majority

34

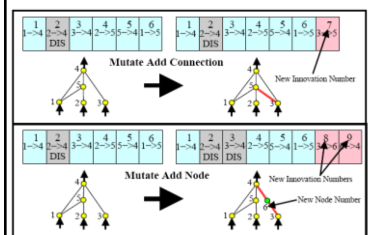
NeuroEvolution of Augmenting Topologies (NEAT) [61,63]

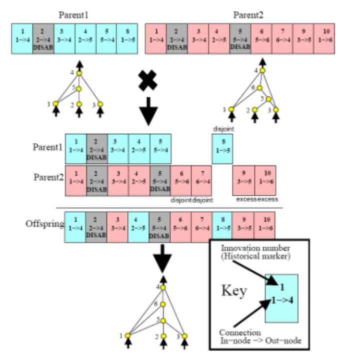
- NEAT addressed the major TWEANN problems:
  - Topology matching problem
  - Loss of innovative structures
  - Initial population topology randomization

35

Historical Marking in NEAT

- Addresses topology-matching problem

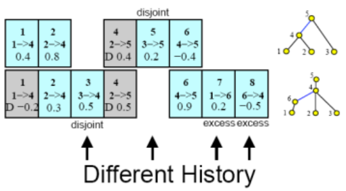


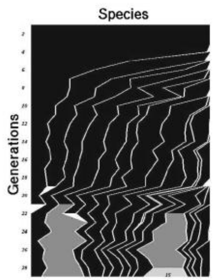


36

Protecting Innovation in NEAT

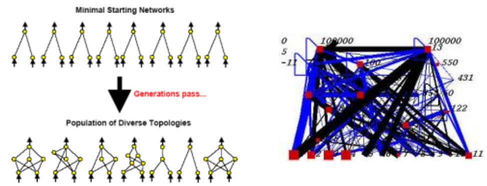
- Addresses loss of innovative structures
- Achieved through *speciation*
  - Individuals compete primarily with others of similar topology





37

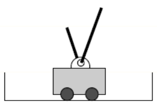
Complexification from Minimal Structure in NEAT

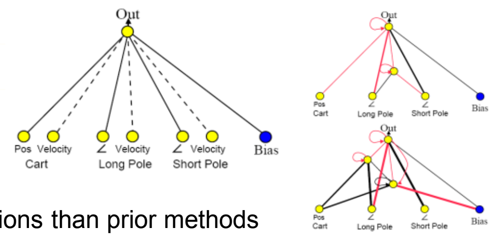


- Addresses initialization problem
- Search begins in minimal-topology space
- Lower-dimensional structures easily optimized
- Useful innovations eventually survive
- So search transitions into good part of higher-dim. space
- *The ticket to high-dimensional space*

38

NEAT Performed Well on Double Pole Balancing Without Velocity Inputs [61]



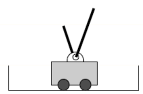


- Fewer evaluations than prior methods
- However, benchmarks can become overemphasized, idiosyncrasies emerge
  - Can be exploited in the future
  - Pole balancing flaws: few or no local optima (tiny population best [57]), no need for hidden nodes

39

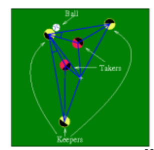
Real Advantages of NEAT

- Unbounded complexity
- Potential for near-minimal solutions
- Diverse topologies and solutions in one run



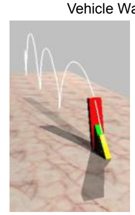
Double Pole Balancing Record [61]

Vehicle Warnings [34]

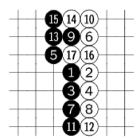


Keepaway Record [66]

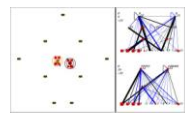
NERO: Real-time Neuroevolution in a Video Game [59]



Hopper [13]



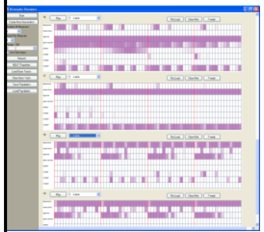
Go [64]



Robot Duel [63]

40

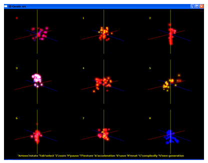
NEAT: Beyond Control and Classification



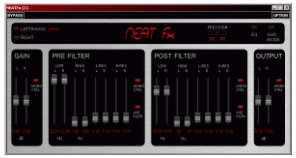
Interactive Drum Pattern Evolution [28]

Harmonic Accompaniment Evolution [29]

Interactive Picture Evolution [52]



Interactive Particle Effect Evolution [26]



Guitar Effect-Pedal Emulation

Dance [12]

41

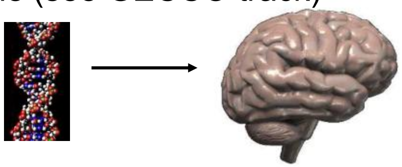
What Makes a Good NE Method?

- Not just about performance
- Also about conceptual foundation
  - Does it open up new possibilities?
  - Can extensions be built upon it?
  - Does it capture something deep from nature?
- Overemphasis on benchmark comparisons obfuscates these critical questions

42

After NEAT: Shift Towards Indirect Encoding

- Also called *Generative and Developmental Systems* (see GECCO track) [3,14,24,39,55,62,75]

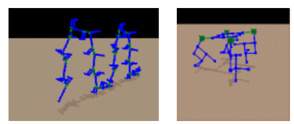


- 100 trillion connections in the human brain
- 30,000 genes in the human genome
- Only possible through highly compressed representation (indirect encoding)

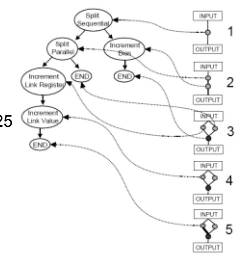
43

Indirectly-Encoded NE (also see GDS Tutorial)

- L-System-based Encoding (body-brain) [30]



- Cellular Encoding (growth program) [24,25]

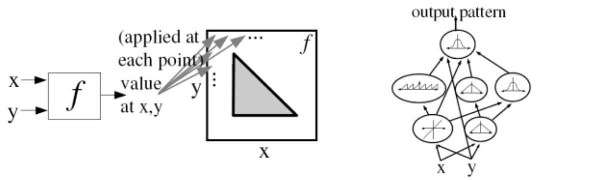


- Analog Genetic Encoding (AGE) [37]
  - Implicit encoding of connection weights in a network

44

An Interesting Observation

- NEAT-evolved networks (called CPPNs [58]) produce nice patterns: Can this ability help to evolve brains?

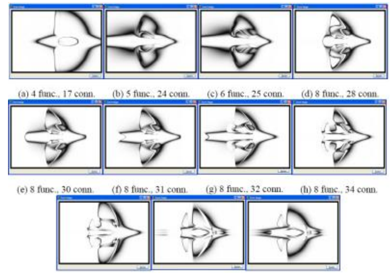


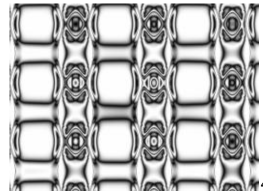
(a) Mapping (b) CPPN = Compositional Pattern Producing Network

45

An Interesting Observation

- NEAT-evolved networks (called CPPNs [58]) produce nice patterns: Can this ability help to evolve brains?



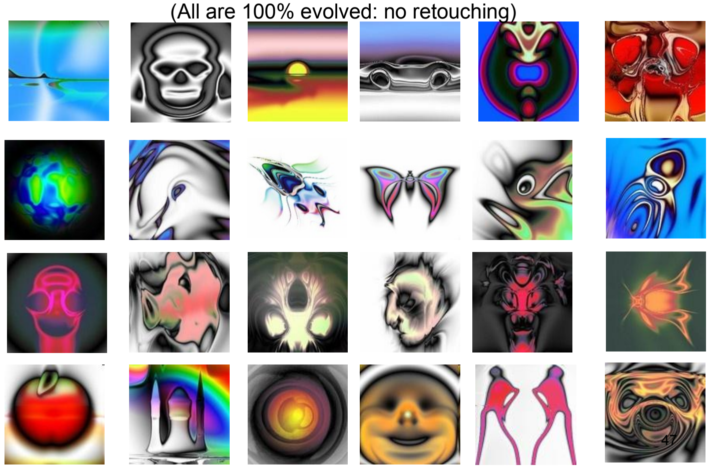


46

CPPN Patterns (Also for brains?)

From http://picbreeder.org [52,53]

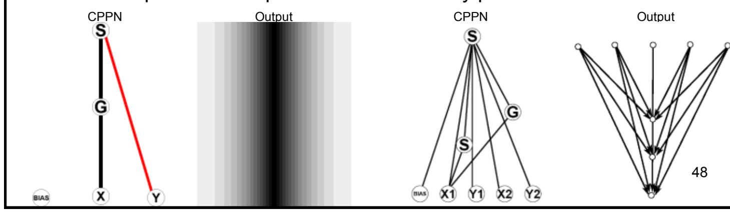
(All are 100% evolved: no retouching)



47

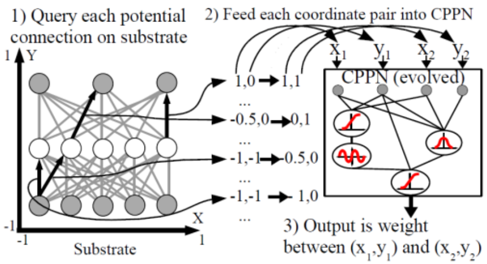
CPPN-based Indirect Encoding: Hypercube-based NEAT (HyperNEAT)[19,60]

- Main insight: 2-D connections *isomorphic* to 4-D points
  - Nodes situated in 2 spatial dimensions (x,y)
  - Connections expressed with 4 spatial dim. (x_1,y_1,x_2,y_2)
- HyperNEAT extends 2-D CPPNs to 4-D
  - CPPN encodes 4-D patterns (i.e. inside a hypercube)
    - 4-D patterns can express the same regularities as 2d patterns
    - 4-D patterns interpreted as connectvitity patterns



48

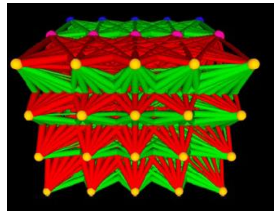
HyperNEAT Indirect Encoding

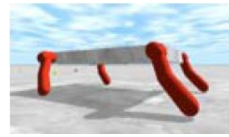


- Substrate is a geometric arrangement of nodes for an ANN which the CPPN can "see"
- The nodes are arranged to exploit the geometry of the problem

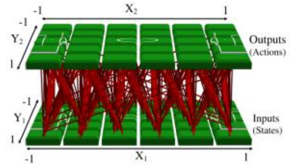
49

Example HyperNEAT Substrates

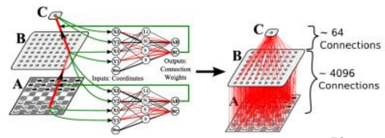




Quadruped Gaits [6,7]



Robocup [70]

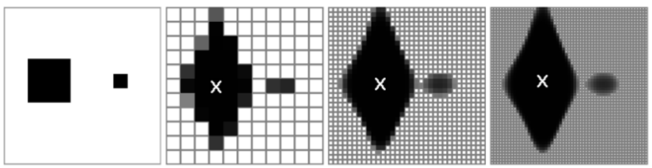


Checkers [18,19]

50

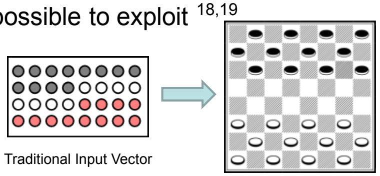
New Capabilities with HyperNEAT

- Evolve very large ANNs (encoded by small CPPNs) [60]



(a) Sensor Field (object placement) (b) 11x11 Target Field (12,827 connections) (c) 33x33 Target Field (1,033,603 connections) (d) 55x55 Target Field (7,970,395 connections)

- Problem geometry now possible to exploit [18,19]
  - e.g. Locality
  - The problem of the torn checkers board

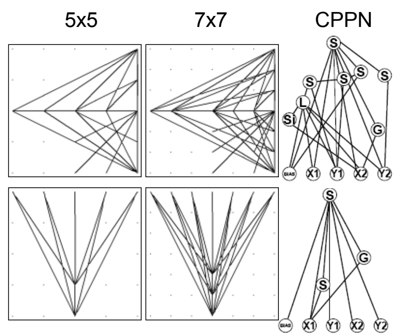


Traditional Input Vector

51

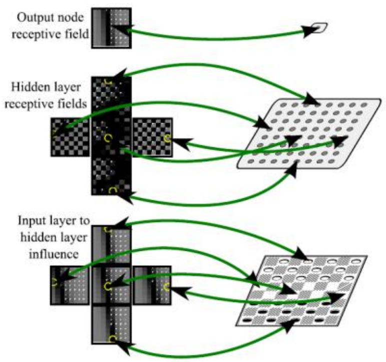
New Capabilities with HyperNEAT

- Scalable connectivity patterns



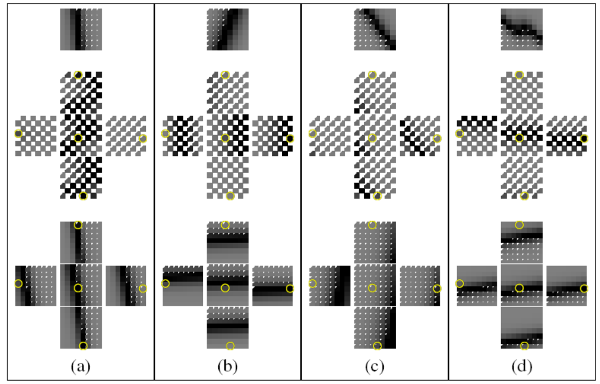
52

Discovery through Geometry: The Geometry of Generality in Checkers [19]



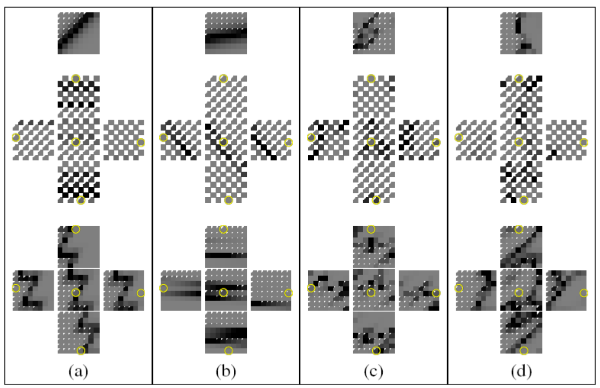
Connectivity of Most General Solutions

Solution	Wins	Losses	Ties
1	53	22	25
2	62	17	21
3	61	17	22
4	54	28	18



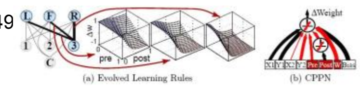
Connectivity of Less General Solutions

Solution	Wins	Losses	Ties
1	35	24	41
2	39	8	53
3	38	33	29
4	35	17	48



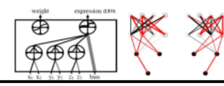
Advanced Issues: More Brain-like ANNs

- Encoding plasticity
  - Long history of approaches [16]
  - Adaptive HyperNEAT [49]



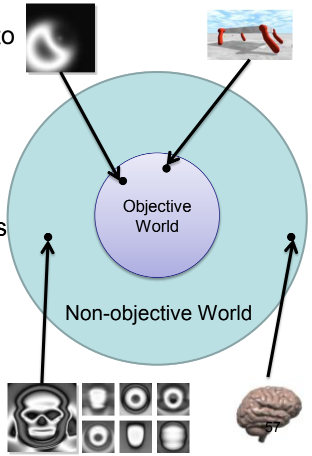
- Evolving node placement in HyperNEAT (ES-HyperNEAT) [50]

- Locality and modularity in HyperNEAT (HyperNEAT-LEO) [71]

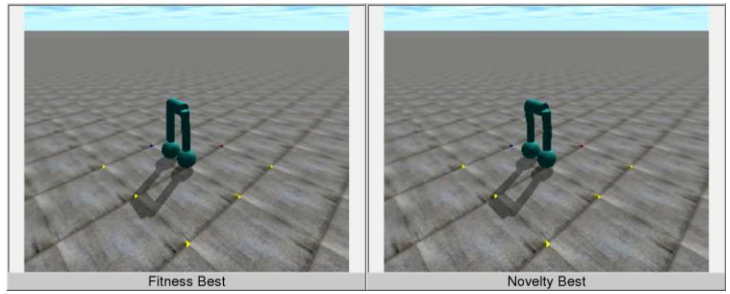


New Insights into Fitness in NE

- Some behaviors may prohibitive to evolve when made objectives
  - (Human intelligence is not a traditional objective in nature)
- Recent research suggests that searching for behavioral novelty can discover behaviors that fail as objectives
  - Novelty search algorithm [35,36]
  - Diversity objectives in multiobjective NE [44]
- Significant implications for research in NE [74]



Novelty and Fitness Bipeds [36]



58

Application Demos

- Driving and collision warning
- Video game applications
- Music
- Multiagent robot control

59

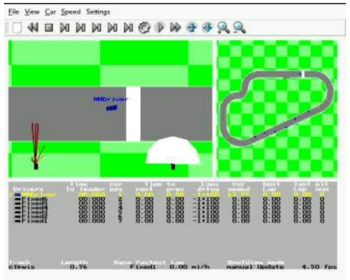
Driving and Collision Warning [34]

- Goal: evolve a collision warning system
  - Looking over the driver's shoulder
  - Adapting to drivers and conditions
  - Collaboration with Toyota

60

Driving (Evolved in RARS)

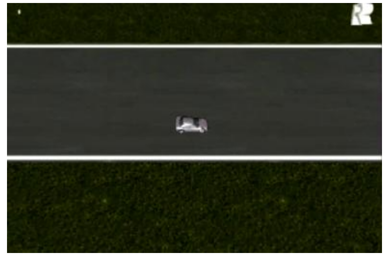
Evolved Alone



Evolved with Obstacles

61

Warning System



In Simulation

In Real Robot

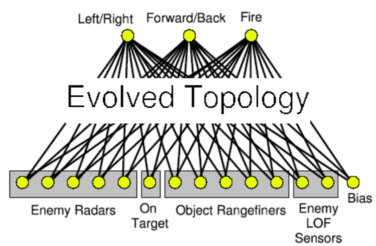
62

Video Games: NeuroEvolving Robotic Operatives [59]

- Robot agents trained for battle by player in real time through real-time NEAT (rtNEAT)

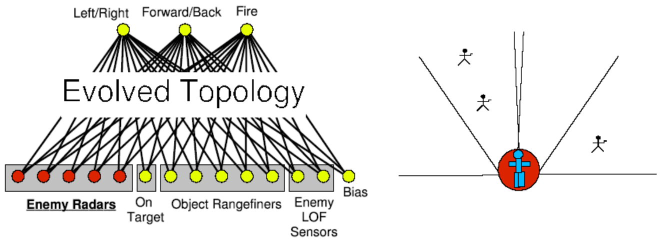
63

NERO Inputs and Outputs



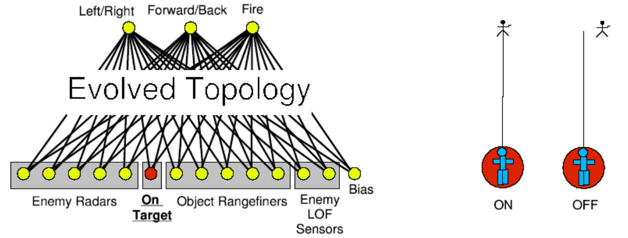
64

Enemy/Friend Radars



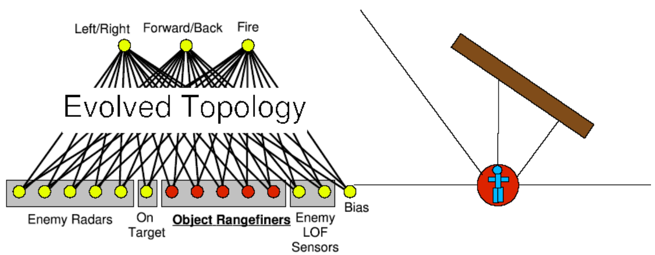
65

Enemy On-Target Sensor



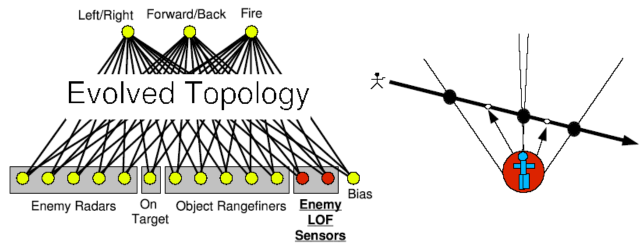
66

Object Rangefinder Sensors



67

Enemy Line-of-Fire Sensors



68

Learning to Approach an Enemy

69

Trained to Solve a Maze

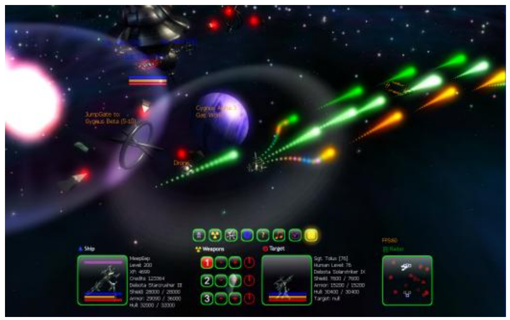
70

Learning to Avoid Fire

71

Another Video Game: Galactic Arms Race (GAR) [27]

- GAR evolves its own weapon systems
  - Based on player behavior



72

GAR Demo

73

Evolving Musical Accompaniment [29]

- MaestroGenesis implements *Functional Scaffolding for Musical Composition* approach

Scarborough Fair

Scarborough Fair with accompaniment

74

Multiagent Robot Control

- Multiagent HyperNEAT [8,9,10]
  - Learns a *set* of brains instead of a single brain
  - Coordinated team behavior entirely invented by evolution

Numerous Other Applications

- Measuring the mass of the top quark [1]
- Art and dance [12,52]
- Theorem proving [11]
- Time-series prediction [38]
- Computer system optimization [21]
- Manufacturing optimization [23]
- Process control optimization [67,68]
- Etc.

76

Big Questions for the Future

- How complex can evolved ANNs become?
- Can evolved ANNs approach or resemble real brains?
  - How should plasticity play a role?
- What is the right selection pressure to encourage complexity?
  - What is the proper role of explicit objectives?
- How should NE methods be judged?
  - When are benchmark comparisons useful?
  - The problem of objective assessment

Conclusion

- Vast potential for further contributions
  - Natural brains are a proof of concept
- Many promising new directions
  - Indirect encoding
  - Non-objective evolution
- Diverse application domains
  - Anything a brain can do an ANN can try to do
- NE is a disruptive AI technology
  - Not only an optimizer

More information

- My Homepage: http://www.cs.ucf.edu/~kstanley
- NEAT Users Group: http://groups.yahoo.com/group/neat
- Evolutionary Complexity Research Group: http://eplex.cs.ucf.edu
- Picbreeder: http://picbreeder.org
- HyperNEAT Information: http://eplex.cs.ucf.edu/hyperNEATpage/HyperNEAT.html
- Email: kstanley@eecs.ucf.edu

References

[1] Aaltonen et al. (over 100 authors) (2009). Measurement of the top quark mass with dilepton events selected using neuroevolution at CDF. Physical Review Letters, 102(15):2001.

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

[3] Angeline, P. J., Saunders, G. M., and Pollack, J. B. (1993). An evolutionary algorithm that constructs recurrent neural networks. IEEE Transactions on Neural Networks, 5:54–65.

[4] Angeline, P. J., Saunders, G. M., and Pollack, J. B. (2000). An evolutionary algorithm that constructs recurrent neural networks. IEEE Transactions on Neural Networks.

[5] Cliff, D., Harvey, I., and Husbands, P. (1993). Explorations in evolutionary robotics. Adaptive Behavior, 2:73–110.

[6] Clune, J., Pennock, R. T., and Ofria, C. (2009). The sensitivity of HyperNEAT to different geomet- ric representations of a problem. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2009). New York, NY, USA: ACM Press.

[7] Clune, J., Stanley, K. O., Pennock, R. T., and Ofria, C. (2011). On the performance of indirect encoding across the continuum of regularity. IEEE Transactions on Evolutionary Computation.

[8] D'Ambrosio, D., Lehman, J., Risi, S., and Stanley, K. O. (2010). Evolving policy geometry for scal- able multiagent learning. In Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2010), 731–738. International Foundation for Autonomous Agents and Multiagent System.

[9] D'Ambrosio, D., Lehman, J., Risi, S., and Stanley, K. O. (2011). Task switching in multiagent learning through indirect encoding. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2011). Piscataway, NJ: IEEE.

[10] D'Ambrosio, D. B., and Stanley, K. O. (2008). Generative encoding for multiagent learning. In Pro- ceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008). New York, NY: ACM Press.

[11] Desai, N. S., and Miikkulainen, R. (2000). Neuro-evolution and natural deduction. In Proceedings of The First IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks, 64–69. Piscataway, NJ: IEEE.

[12] Dubbin, G., and Stanley, K. O. (2010). Learning to dance through interactive evolution. In Proceed- ings of the Eigth European Event on Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMUSART 2010). New York, NY: Springer.

[13] Fagerlund, M. (2003–2006). DelphiNEAT homepage. http://www.cambrianlabs.com/mattias/DelphiNEAT/. [14] Floreano, D., Du¨ rr, P., and Mattiussi, C. (2008). Neuroevolution: from architectures to learning. Evolu- tionary Intelligence, 1:47–62.

[15] Floreano, D., and Mondada, F. (1998). Evolutionary neurocontrollers for autonomous mobile robots. Neural Networks, 11:1461–1478.

[16] Floreano, D., and Urzelai, J. (2000). Evolutionary robots with on-line self-organization and behavioral fitness. Neural Networks, 13:431–4434.

[17] Fullmer, B., and Miikkulainen, R. (1992). Using marker-based genetic encoding of neural networks to evolve finite-state behaviour. In Varela, F. J., and Bourgine, P., editors, Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, 255–262. Cambridge, MA: MIT Press.

[18] Gauci, J., and Stanley, K. O. (2008). A case study on the critical role of geometric regularity in machine learning. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-2008). Menlo Park, CA: AAAI Press.

[19] Gauci, J., and Stanley, K. O. (2010). Autonomous evolution of topographic regularities in artificial neural networks. Neural Computation, 22(7):1860–1898.

[20] Gomez, F. (2003). Robust Non-Linear Control Through Neuroevolution. PhD thesis, Department of Com- puter Sciences, The University of Texas at Austin.

[21] Gomez, F., Burger, D., and Miikkulainen, R. (2001). A neuroevolution method for dynamic resource allocation on a chip multiprocessor. In Proceedings of the INNS-IEEE International Joint Conference on Neural Networks, 2355–2361. Piscataway, NJ: IEEE.

[22] Gomez, F., and Miikkulainen, R. (2003). Active guidance for a finless rocket using neuroevolution. In Proceedings of the Genetic and Evolutionary Computation Conference, 2084–2095. San Francisco: Kaufmann.

[23] Greer, B., Hakonen, H., Lahdelma, R., and Miikkulainen, R. (2002). Numerical optimization with neuroevolution. In Proceedings of the 2002 Congress on Evolutionary Computation, 361–401. Piscataway, NJ: IEEE.

[24] Gruau, F., and Whitley, D. (1993). Adding learning to the cellular development of neural networks: Evolution and the Baldwin effect. Evolutionary Computation, 1:213–233.

[25] Gruau, F., Whitley, D., and Pyeatt, L. (1996). A comparison between cellular encoding and direct encoding for genetic neural networks. In Koza, J. R., Goldberg, D. E., Fogel, D. B., and Riolo, R. L., editors, Genetic Programming 1996: Proceedings of the First Annual Conference, 81–89. Cambridge, MA: MIT Press.

[26] Hastings, E., Guha, R., and Stanley, K. O. (2007). Neat particles: Design, representation, and animation of particle system effects. In Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG-07). Piscataway, NJ: IEEE Press.

[27] Hastings, E. J., Guha, R. K., and Stanley, K. O. (2010). Automatic content generation in the galactic arms race video game. IEEE Transactions on Computational Intelligence and AI in Games, 1(4):245–263.

[28] Hoover, A. K., and Stanley, K. O. (2009). Exploiting functional relationships in musical composition. Connection Science Special Issue on Music, Brain, and Cognition, 21(2 and 3):227–251.

[29] Hoover, A. K., Szerlip, P. A., and Stanley, K. O. (2011). Interactively evolving harmonies through functional scaffolding. In GECCO '11: Proceedings of the 13th annual conference on Genetic and evolutionary computation, 387–394. Dublin, Ireland: ACM.

[30] Hornby, G. S., and Pollack, J. B. (2002). Creating high-level components with a generative representa- tion for body-brain evolution. Artificial Life, 8(3).

[31] Hornby, G. S., Takamura, S., Yokono, J., Hanagata, O., Fujita, M., and Pollack, J. (2000). Evolution of controllers from a high-level simulator to a high DOF robot. In Evolvable Systems: From Biology to Hardware; Proceedings of the Third International Conference, 80–89. Berlin: Springer.

[32] Igel, C. (2003). Neuroevolution for reinforcement learning using evolution strategies. In Sarker, R., Reynolds, R., Abbass, H., Tan, K. C., McKay, B., Essam, D., and Gedeon, T., editors, Proceedings of the 2003 Congress on Evolutionary Computation, 2588–2595. Piscataway, NJ: IEEE Press.

[33] James, D., and Tucker, P. (2005). Evolving a neural network active vision system for shape discrimina- tion. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005) Late Breaking Papers. New York, NY: ACM Press.

[34] Kohl, N., Stanley, K., Miikkulainen, R., Samples, M., and Sherony, R. (2006). Evolving a real-world vehicle warning system. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006), 1681–1688.

[35] Lehman, J., and Stanley, K. O. (2008). Exploiting open-endedness to solve problems through the search for novelty. In Bullock, S., Noble, J., Watson, R., and Bedau, M., editors, Proceedings of the Eleventh International Conference on Artificial Life (Alife XI). Cambridge, MA: MIT Press.

[36] Lehman, J., and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. Evolutionary Computation, 19(2):189–223.

[37] Mattiussi, C., and Floreano, D. (2006). Analog Genetic Encoding for the Evolution of Circuits and Networks. IEEE Transactions on Evolutionary Computation, 11(5):596–607.

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

[39] Mjolsness, E., Sharp, D. H., and Alpert, B. K. (1989). Scaling, machine learning, and genetic neural nets. Advances in Applied Mathematics, 10:137–163.

[40] Montana, D. J., and Davis, L. (1989). Training feedforward neural networks using genetic algorithms. In Proceedings of the 11th International Joint Conference on Artificial Intelligence, 762–767. San Francisco: Kaufmann.

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

[42] Moriarty, D. E., and Miikkulainen, R. (1996). Evolving obstacle avoidance behavior in a robot arm. In Maes, P., Mataric, M. J., Meyer, J.-A., Pollack, J., and Wilson, S. W., editors, From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, 468–475. Cambridge, MA: MIT Press.

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

[44] Mouret, J.-B., and Doncieux, S. (2009). Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC-2009, 1161–1168. IEEE.

[45] Nolfi, S., and Floreano, D. (2000). Evolutionary Robotics. Cambridge: MIT Press.

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

[47] Pujol, J. C. F., and Poli, R. (1998). Evolving the topology and the weights of neural networks using a dual representation. Applied Intelligence Journal, 8(1):73–84. Special Issue on Evolutionary Learning.

[48] Radcliffe, N. J. (1993). Genetic set recombination and its application to neural network topology opti- mization. Neural computing and applications, 1(1):67–90.

[49] Risi, S., and Stanley, K. O. (2010). Indirectly encoding neural plasticity as a pattern of local rules. In Proceedings of the 11th International Conference on Simulation of Adaptive Behavior (SAB2010). Berlin: Springer.

[50] Risi, S., and Stanley, K. O. (2011). Enhancing ES-HyperNEAT to evolve more complex regular neural networks. In GECCO '11: Proceedings of the 13th annual conference on Genetic and evolutionary computation, 1539–1546. Dublin, Ireland: ACM.

[51] Schaffer, J. D., Whitley, D., and Eshelman, L. J. (1992). Combinations of genetic algorithms and neural networks: A survey of the state of the art. In Whitley, D., and Schaffer, J., editors, Proceedings of the International Workshop on Combinations of Genetic Algorithms and Neural Networks, 1–37. Los Alamitos, CA: IEEE Computer Society Press.

[52] Secretan, J., Beato, N., D.Ambrosio, D. B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J. T., and Stanley, K. O. (2011). Picbreeder: A case study in collaborative evolutionary exploration of design space. Evolutionary Computation, 19(3):345–371.

[53] Secretan, J., Beato, N., D'Ambrosio, D. B., Rodriguez, A., Campbell, A., and Stanley, K. O. (2008). Picbreeder: Evolving pictures collaboratively online. In CHI '08: Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, 1759–1768. New York, NY, USA: ACM.

[54] Seys, C. W., and Beer, R. D. (2004). Evolving walking: The anatomy of an evolutionary search. In Schaal, S., Ijspeert, A., Billard, A., Vijayakumar, S., Hallam, J., and Meyer, J.-A., editors, From Animals to Animats 8: Proceedings of the Eight International Conference on Simulation of Adaptive Behavior, 357–363. Cambridge, MA: MIT Press.

[55] Siddiqi, A. A., and Lucas, S. M. (1998). A comparison of matrix rewriting versus direct encoding for evolving neural networks. In Proceedings of IEEE International Conference on Evolutionary Computation, 392–397. Piscataway, NJ: IEEE.

[56] Sit, Y. F., and Miikkulainen, R. (2005). Learning basic navigation for personal satellite assistant using neuroevolution. In Proceedings of the Genetic and Evolutionary Computation Conference.

[57] Stanley, K. O. (2003). Efficient Evolution of Neural Networks Through Complexification. PhD thesis, De- partment of Computer Sciences, The University of Texas at Austin, Austin, TX.

[58] Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. Genetic Programming and Evolvable Machines Special Issue on Developmental Systems, 8(2):131–162.

[59] Stanley, K. O., Bryant, B. D., and Miikkulainen, R. (2005). Real-time neuroevolution in the NERO video game. IEEE Transactions on Evolutionary Computation Special Issue on Evolutionary Computation and Games, 9(6):653–668.

[60] Stanley, K. O., D'Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based indirect encoding for evolving large-scale neural networks. Artificial Life, 15(2):185–212.

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

[62] Stanley, K. O., and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. Artificial Life, 9(2):93–130.

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

[64] Stanley, K. O., and Miikkulainen, R. (2004). Evolving a roving eye for Go. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004). Berlin: Springer Verlag.

[65] Sutton, R. S., and Barto, A. G. (1998). Reinforcement Learning: An Introduction. Cambridge, MA: MIT Press.

[66] Taylor, M. E., Whiteson, S., and Stone, P. (2006). Comparing evolutionary and temporal difference methods in a reinforcement learning domain. In GECCO 2006: Proceedings of the Genetic and Evolution- ary Computation Conference, 1321–1328.

85

[67] v. E. Conradie, A., Miikkulainen, R., and Aldrich, C. (2002). Adaptive control utilising neural swarm- ing. In Proceedings of the Genetic and Evolutionary Computation Conference. San Francisco: Kaufmann.

[68] v. E. Conradie, A., Miikkulainen, R., and Aldrich, C. (2002). Intelligent process control utilizing sym- biotic memetic neuro-evolution. In Proceedings of the 2002 Congress on Evolutionary Computation.

[69] Valsalam, V. K., and Miikkulainen, R. (2008). Modular neuroevolution for multilegged locomotion. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008). New York, NY: ACM Press.

[70] Verbancsics, P., and Stanley, K. O. (2010). Evolving static representations for task transfer. Journal of Machine Learning Research (JMLR), 11:1737–1769.

[71] Verbancsics, P., and Stanley, K. O. (2011). Constraining connectivity to encourage modularity in Hyper- NEAT. In GECCO '11: Proceedings of the 13th annual conference on Genetic and evolutionary computation, 1483–1490. Dublin, Ireland: ACM.

[72] Whitley, D., Dominic, S., Das, R., and Anderson, C. W. (1993). Genetic reinforcement learning for neurocontrol problems. Machine Learning, 13:259–284.

[73] Wieland, A. P. (1990). Evolving controls for unstable systems. In Touretzky, D. S., Elman, J. L., Se- jnowski, T. J., and Hinton, G. E., editors, Connectionist Models: Proceedings of the 1990 Summer School, 91–102. San Francisco: Kaufmann.

[74] Woolley, B. G., and Stanley, K. O. (2011). On the deleterious effects of a priori objectives on evolution and representation. In GECCO '11: Proceedings of the 13th annual conference on Genetic and evolutionary computation, 957–964. Dublin, Ireland: ACM.

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

[76] Zhang, B.-T., and Muhlenbein, H. (1993). Evolving optimal neural networks using genetic algorithms with Occam's razor. Complex Systems, 7:199–220.

86