Evolving Neural Networks

Risto Miikkulainen

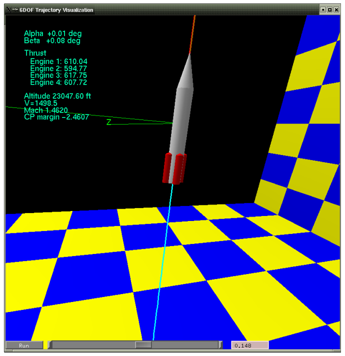
Department of Computer Science

The University of Texas at Austin

http://www.cs.utexas.edu/∼risto

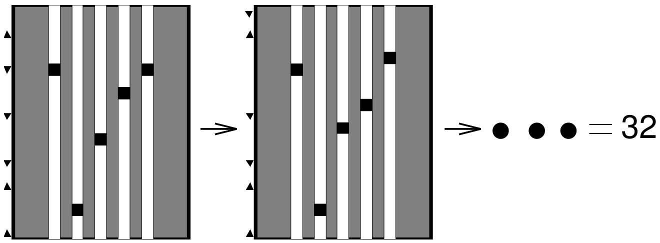
CO'11, July 12–16, 2011, Dublin, Ireland.
1 978-1-4503-0690-4/11/07.

Why Neuroevolution?



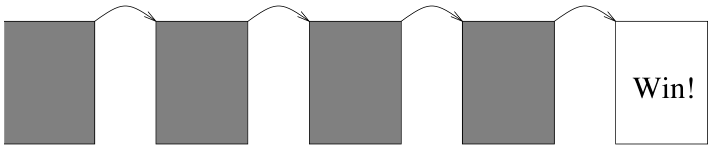
- Neural nets powerful in many statistical domains
  - E.g. control, pattern recognition, prediction, decision making
  - Where no good theory of the domain exists
- Good supervised training algorithms exist
  - Learn a nonlinear function that matches the examples
- What if correct outputs are not known?

Sequential Decision Tasks



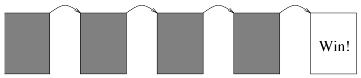
- POMDP: Sequence of decisions creates a sequence of states
- No targets: Performance evaluated after several decisions
- Many important real-world domains:
  - Robot/vehicle/traffic control
  - Computer/manufacturing/process optimization
  - Game playing

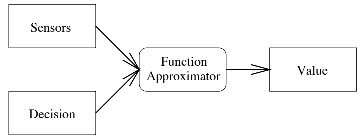
Forming Decision Strategies



- Traditionally designed by hand
  - Too complex: Hard to anticipate all scenarios
  - Too inflexible: Cannot adapt on-line
- Need to discover through exploration
  - Based on sparse reinforcement
  - Associate actions with outcomes

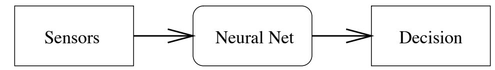
Standard Reinforcement Learning





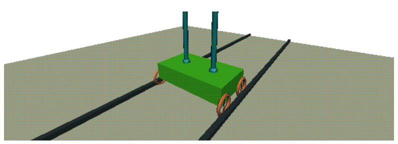
- AHC, Q-learning, Temporal Differences
  - Generate targets through prediction errors
  - Learn when successive predictions differ
- Predictions represented as a value function
  - Values of alternatives at each state
- Difficult with large/continuous state and action spaces
- Difficult with hidden states

leuroevolution (NE) Reinforcement Learnin



- NE = constructing neural networks with evolutionary algorithms
- Direct nonlinear mapping from sensors to actions
- Large/continuous states and actions easy
  - Generalization in neural networks
- Hidden states disambiguated through memory
  - Recurrency in neural networks [76]

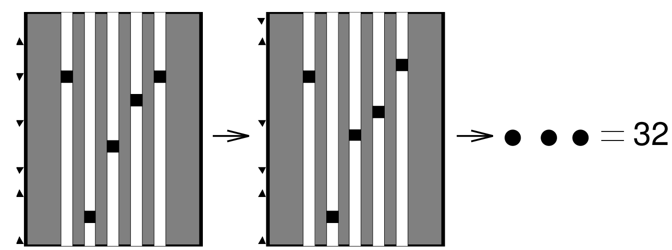
How well does it work?



Poles	Method	Evals	Succ.
One	VAPS	(500,000)	0%
	SARSA	13,562	59%
	Q-MLP	11,331	
	NE	127	
Two	NE	3,416	

- Difficult RL benchmark: Non-Markov Pole Balancing
- NE 3 orders of magnitude faster than standard RL [27]
- NE can solve harder problems

Role of Neuroevolution



- Powerful method for sequential decision tasks [17;27;51;89]
  - Optimizing existing tasks
  - Discovering novel solutions
  - Making new applications possible
- Also may be useful in supervised tasks [47;56]
  - Especially when network topology important
- Unique model of biological adaptation and development [52;61;84]

Outline

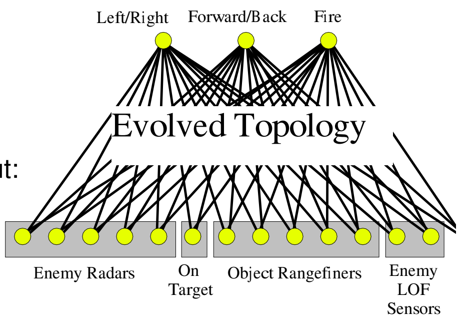
- Basic neuroevolution techniques
- Advanced techniques
  - E.g. combining learning and evolution
- Extensions to applications
- Application examples
  - Control, Robotics, Artificial Life, Games

Neuroevolution Decision Strategies

Input variables describe the state

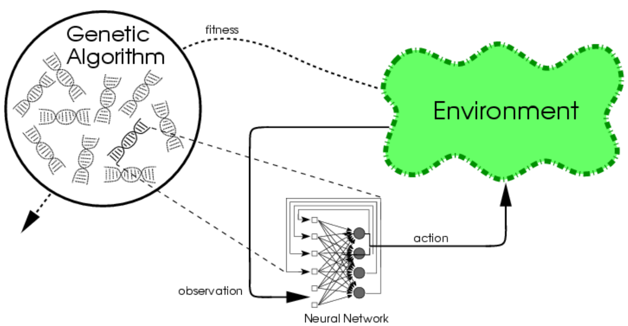
Output variables describe actions

Network between input and output:
- Nonlinear hidden nodes
- Weighted connections



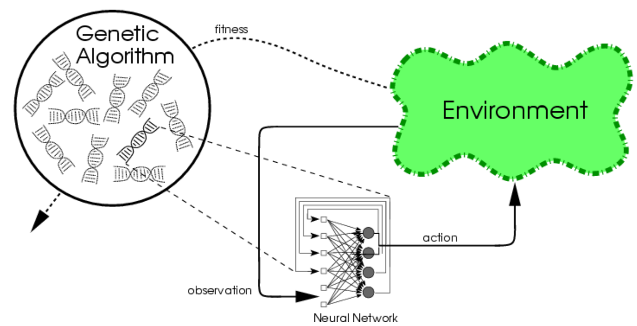
Execution:
- Numerical activation of input
- Performs a nonlinear mapping
- Memory in recurrent connections

Conventional Neuroevolution (CNE)



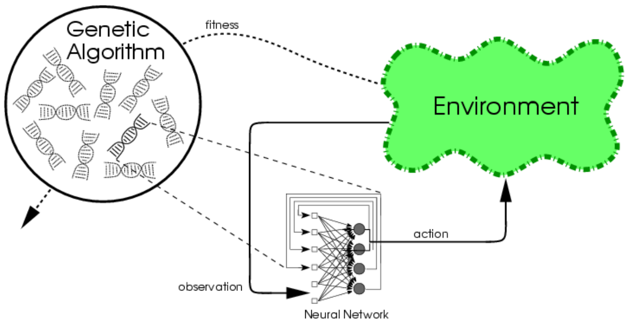
- Evolving connection weights in a population of networks [47;62;89;90]
- Chromosomes are strings of connection weights (bits or real)
  - E.g. 10010110101100101111001
  - Usually fully connected, fixed topology
  - Initially random

Conventional Neuroevolution (2)



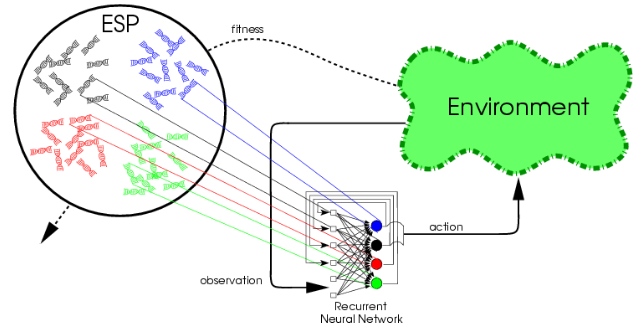
- Parallel search for a solution network
  - Each NN evaluated in the task
  - Good NN reproduce through crossover, mutation
  - Bad thrown away
- Natural mapping between genotype and phenotype
  - GA and NN are a good match!

Problems with CNE



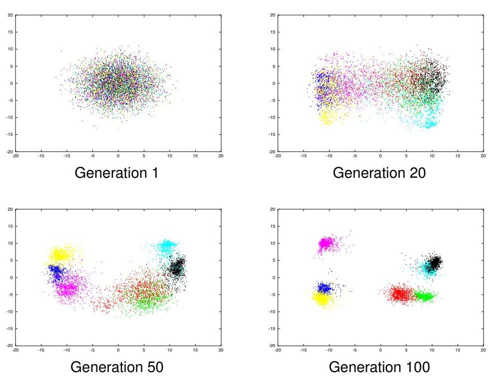
- Evolution converges the population (as usual with EAs)
  - Diversity is lost; progress stagnates
- Competing conventions
  - Different, incompatible encodings for the same solution
- Too many parameters to be optimized simultaneously
  - Thousands of weight values at once

Advanced NE 1: Evolving Partial Networks



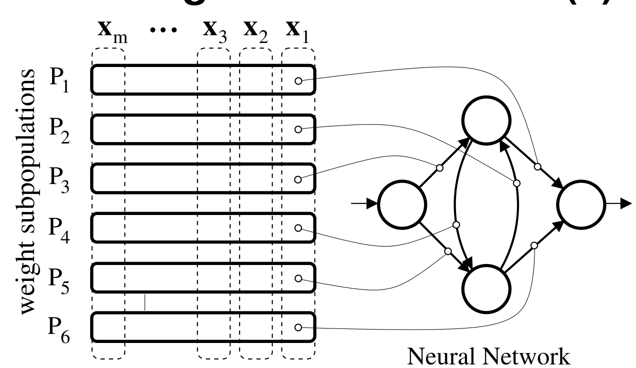
- Evolving individual neurons to cooperate in networks [1;50;56]
- E.g. Enforced Sub-Populations (ESP [23])
  - Each (hidden) neuron in a separate subpopulation
  - Fully connected; weights of each neuron evolved
  - Populations learn compatible subtasks

Evolving Neurons with ESP



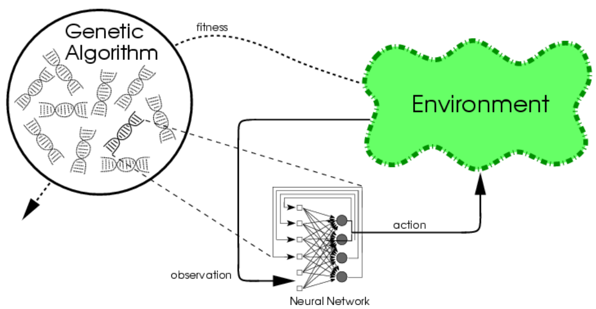
- Evolution encourages diversity automatically
  - Good networks require different kinds of neurons
- Evolution discourages competing conventions
  - Neurons optimized for compatible roles
- Large search space divided into subtasks
  - Optimize compatible neurons

Evolving Partial Networks (2)



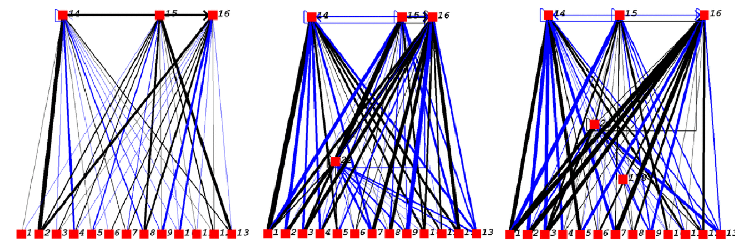
- Extend the idea to evolving connection weights
- E.g. Cooperative Synapse NeuroEvolution (CoSyNE [27])
  - Connection weights in separate subpopulations
  - Networks formed by combining neurons with the same index
  - Networks mutated and recombined; indices permutated
- Sustains diversity, results in efficient search

Advanced NE 2: Evolutionary Strategies



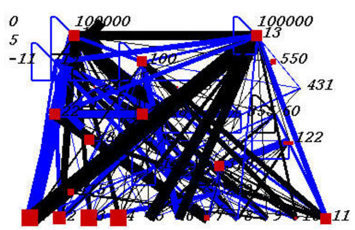
- Evolving complete networks with ES (CMA-ES[34])
- Small populations, no crossover
- Instead, intelligent mutations
  - Adapt covariance matrix of mutation distribution
  - Take into account correlations between weights
- Smaller space, less convergence, fewer conventions

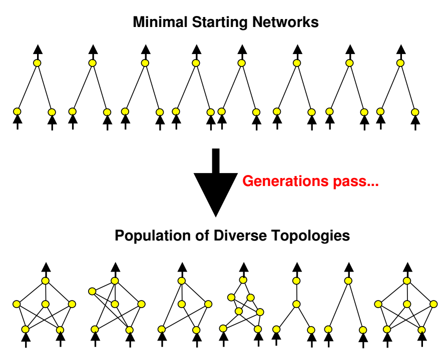
Advanced NE 3: Evolving Topologies



- Optimizing connection weights and network topology[3;17;21;91]
- E.g. Neuroevolution of Augmenting Topologies (NEAT[69;71])
- Based on *Complexification*
- Of networks:
  - Mutations to add nodes and connections
- Of behavior:
  - Elaborates on earlier behaviors

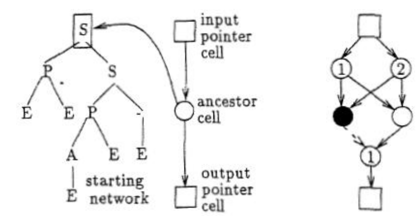
Why Complexification?





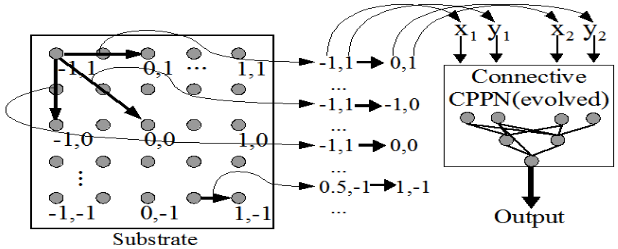
- Problem with NE: Search space is too large
- Complexification keeps the search tractable
  - Start simple, add more sophistication
- Incremental construction of intelligent agents

Advanced NE 4: Indirect Encodings

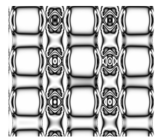


- Instructions for constructing the network evolved
  - Instead of specifying each unit and connection[3;17;46;67;91]
- E.g. Cellular Encoding (CE[29])
- Grammar tree describes construction
  - Sequential and parallel cell division
  - Changing thresholds, weights
  - A “developmental” process that results in a network

Indirect Encodings (2)

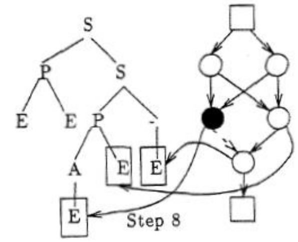


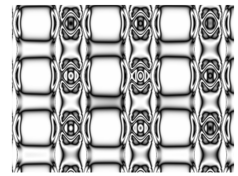
- Encode the networks as spatial patterns
- E.g. Hypercube-based NEAT (HyperNEAT[12])
- Evolve a neural network (CPPN) to generate spatial patterns
  - 2D CPPN: (x, y) input $\longrightarrow$ grayscale output
  - 4D CPPN: (x_1, y_1, x_2, y_2) input $\longrightarrow$ w output
  - Connectivity and weights can be evolved indirectly
  - Works with very large networks (millions of connections)



21/64

Properties of Indirect Encodings

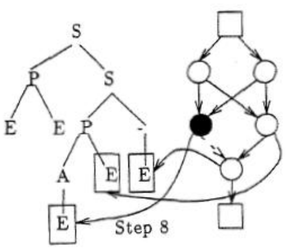


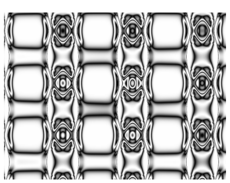


- Smaller search space
- Avoids competing conventions
- Describes classes of networks efficiently
- Modularity, reuse of structures
  - Recurrency symbol in CE: XOR $\longrightarrow$ pari
  - Repetition with variation in CPPNs
  - Useful for evolving morphology

22/64

Properties of Indirect Encodings

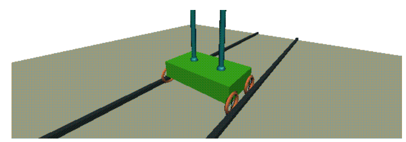




- Not fully explored (yet)
  - See e.g. GDS track at GECCO
- Promising current work
  - More general L-systems; developmental codings; embryogeny[72]
  - Scaling up spatial coding[13;22]
  - Genetic Regulatory Networks[57]
  - Evolution of symmetries[79]

23/64

How Do the NE Methods Compare?



Poles	Method	Evals
Two	CE	(840,000)
	CNE	87,623
	ESP	26,342
	NEAT	6,929
	CMA-ES	6,061
	CoSyNE	3,416

Two poles, no velocities, damping fitness[27]

- Advanced methods better than CNE
- Advanced methods still under development
- Indirect encodings future work

24/64

Further NE Techniques

- Incremental, multiobjective, novelty evolution [40;64 25;38;39;64;78;90]
- Utilizing population culture [5;44]
- Evolving NN ensembles and modules [41;55;58;86]
- Evolving transfer functions and learning rules [8;60;75]
- Evolving value functions [87]
- Combining learning and evolution

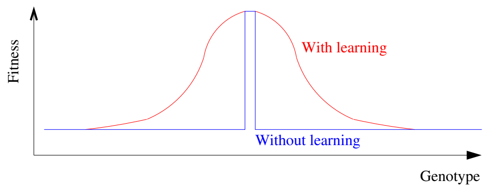
25/64

Combining Learning and Evolution

- Good learning algorithms exist for NN
  - Why not use them as well?
- Evolution provides structure and initial weights
- Fine tune the weights by learning
- Lamarckian evolution is possible
  - Coding weight changes back to chromosome
- Difficult to make it work
  - Diversity reduced; progress stagnates

26/64

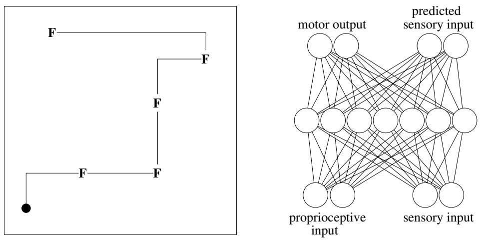
Baldwin Effect



- Learning can guide Darwinian evolution [4;31]
  - Makes fitness evaluations more accurate
- With learning, more likely to find the optimum if close
- Can select between good and bad individuals better
  - Lamarckian not necessary
- How can we implement it?
  - How to obtain training targets?

27/64

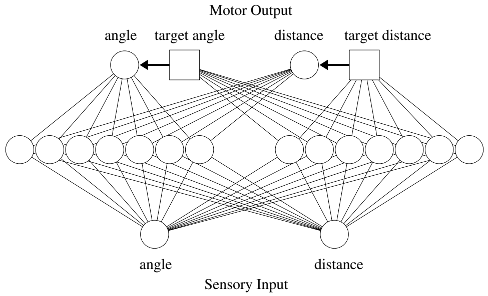
Targets from a Related Task



- Learning in a related task is sufficient
- E.g. foraging for food in a microworld [52]
  - Network sees the state, outputs motor commands
  - Trained with backprop to predict the next input
  - Training emphasizes useful hidden-layer representations
  - Allows more accurate evaluations

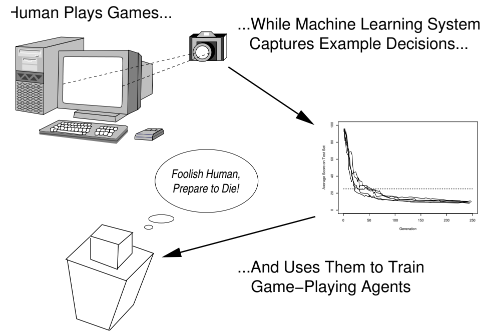
28/64

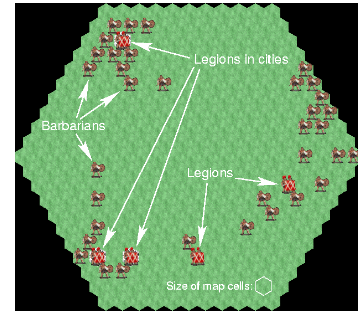
Evolving the Targets



- Evolve extra outputs to provide targets
- E.g. in the foraging task[54]
  - Motor outputs and targets with separate hidden layers
  - Motor weights trained with backprop, targets evolved
  - Targets do not correspond to optimal performance: Direct system towards useful learning experiences

Targets from Humans





- Humans can demonstrate desired behavior
- E.g. fine tuning game agents[7]
  - Human observer identifies suboptimal behavior
  - Drives the NPC with a joystick
  - Agent placed in the same input situation
  - Backpropagate from human actions

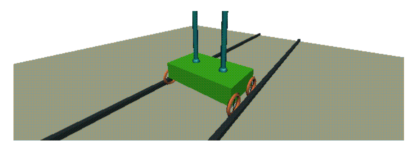
Extending NE to Applications

- Control
- Robotics
- Artificial life
- Gaming

Issues:

- Combining evolution with human knowledge[7;16;93] (Karpov GECCO'11)
- Making evolution run in real-time[70]
- Utilizing coevolution[59;73]
- Utilizing problem symmetry and hierarchy[36;79;81]
- Evolving multimodal behavior[63;64;86]
- Evolving teams of agents[6;70;92]

Applications to Control



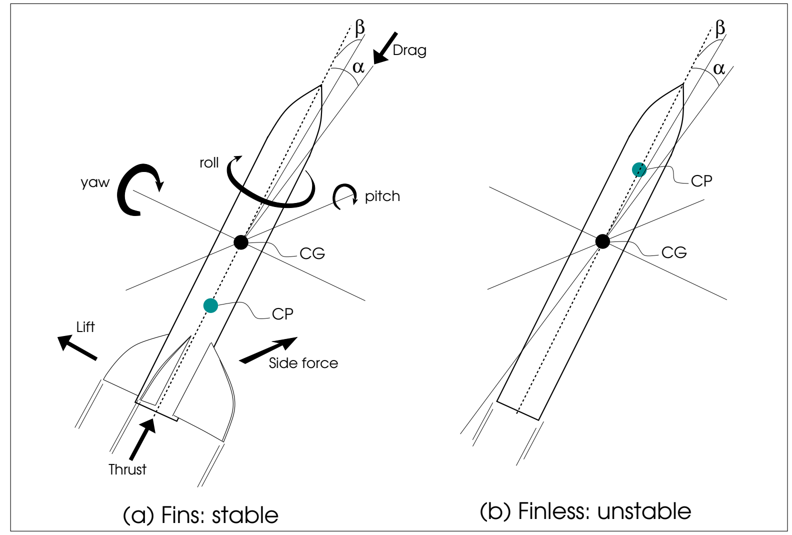
- Pole-balancing benchmark
  - Originates from the 1960s
  - Original 1-pole version too easy
  - Several extensions: acrobat, jointed, 2-pole, particle chasing[55]
- Good surrogate for other control tasks
  - Vehicles and other physical devices
  - Process control[82]

Controlling a Finless Rocket

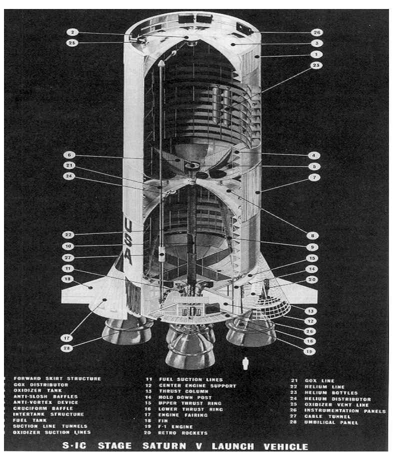
Task: Stabilize a finless version of the Interorbital Systems RSX-2 sounding rocket [26]

- Scientific measurements in the upper atmosphere
- 4 liquid-fueled engines with variable thrust
- Without fins will fly much higher for same amount of fuel

Rocket Stability

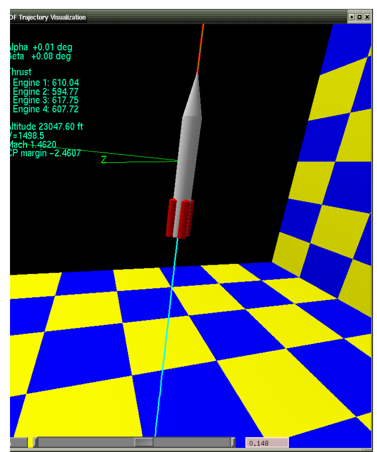


Active Rocket Guidance



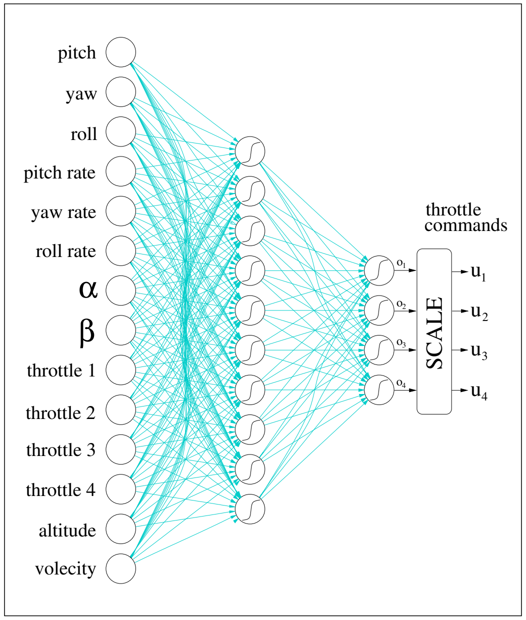
- Used on large scale launch vehicles (Saturn, Titan)
- Typically based on classical linear feedback control
- High level of domain knowledge require
- Expensive, heavy

Simulation Environment: JSBSim

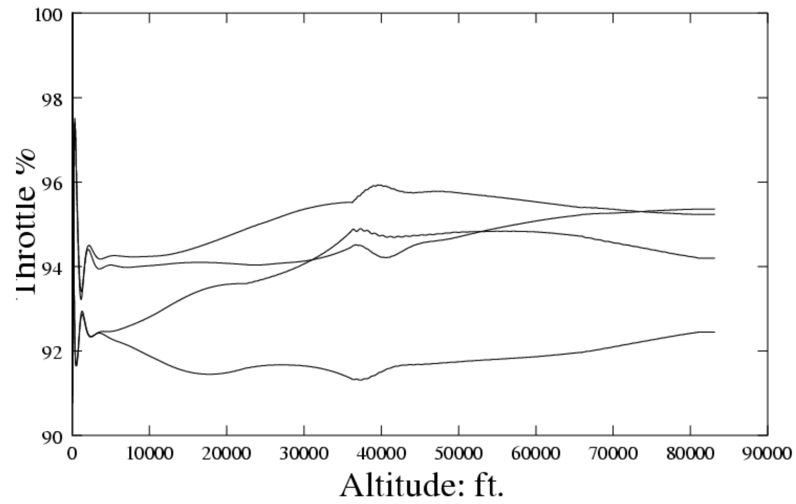


- General rocket simulator
- Models complex interaction between air frame, propulsion, aerodynamics, and atmosphere
- Used by IOS in testing their rocket design
- Accurate geometric model of the RSX-2

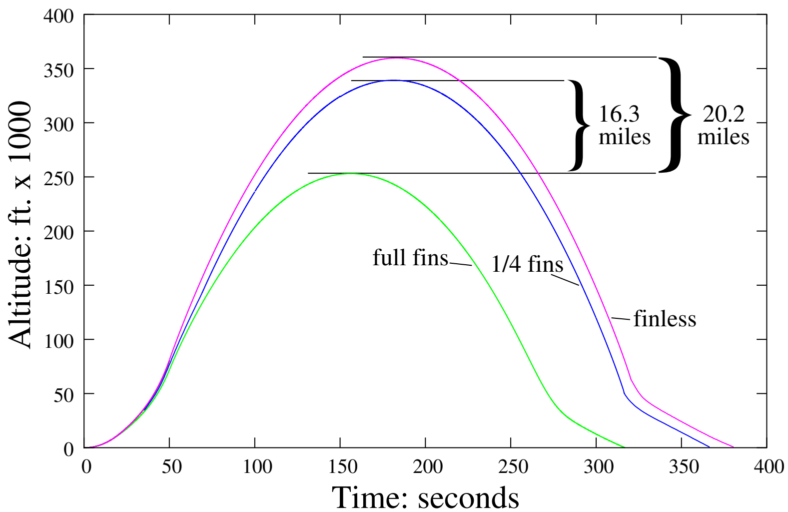
Rocket Guidance Network



Results: Control Policy



Results: Apogee



- DEMO

Applications to Robotics

Controlling a robot arm [49]

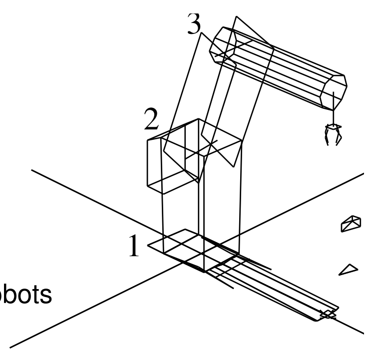
- Compensates for an inop motor

Robot walking [33;66;81]

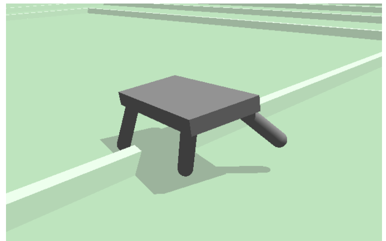
- Various physical platforms

Mobile robots [11;18;53;68]

- Transfers from simulation to physical robots
- Evolution possible on physical robots

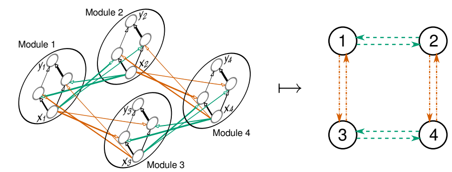


Multilegged Walking



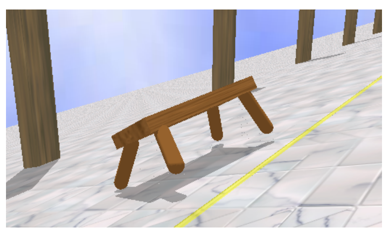
- Navigate rugged terrain better than wheeled robots
- Controller design is more challenging
  - Leg coordination, robustness, stability, fault-tolerance, ...
- Hand-design is generally difficult and brittle
- Large design space often makes evolution ineffective

ENSO: Symmetry Evolution Approach



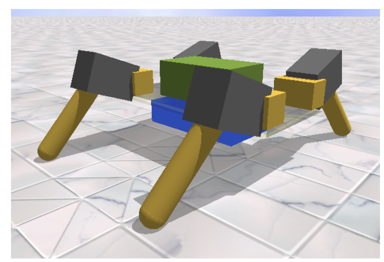
- Symmetry evolution approach [79;80;81]
  - A neural network controls each leg
  - Connections between controllers evolved through symmetry breaking
  - Connections withing individual controllers evolved through neuroevolution

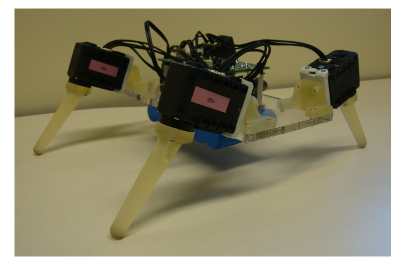
Robust, Effective Solutions



- Different gaits on flat ground
  - Pronk, pace, bound, trot
  - Changes gait to get over obstacles
- Asymmetric gait on inclines
  - One leg pushes up, others forward
  - Hard to design by hand
- DEMO

Transfer to a Physical Robot



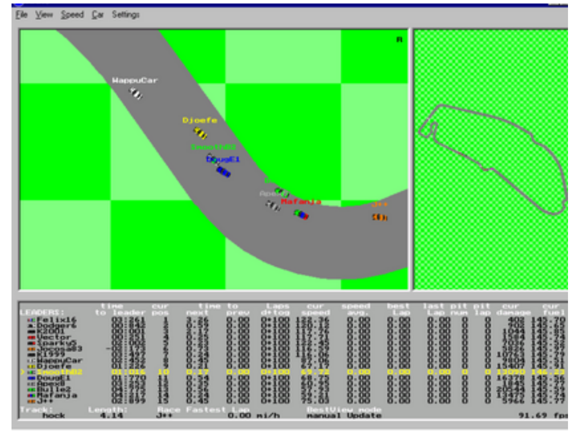


- Built at Hod Lipson's lab (Cornell U.)
  - Standard motors, battery, controller board
  - Custom 3D-printed legs, attachments
  - Simulation modified to match
- General, robust transfer
  - Noise to actuators during simulation
  - Generalizes to different surfaces, motor speeds
  - Evolved a solution for 3-legged walking!
- DEMO

Driving and Collision Warning

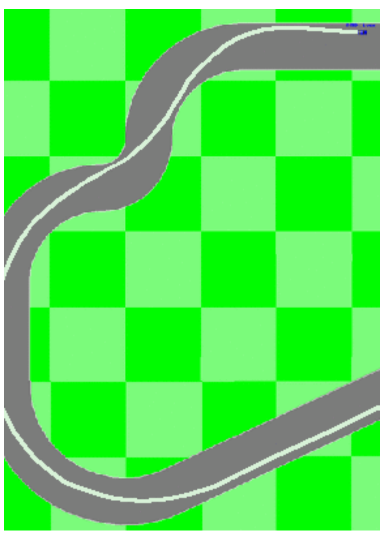
- Goal: evolve a collision warning system
  - Looking over the driver's shoulder
  - Adapting to drivers and conditions
  - Collaboration with Toyota [37]

The RARS Domain



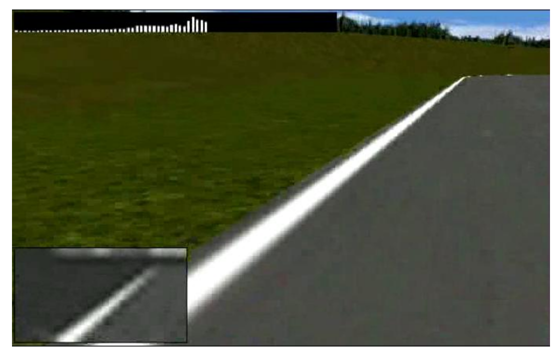
- RARS: Robot Auto Racing Simulator
  - Internet racing community
  - Hand-designed cars and drivers
  - First step towards real traffic

Evolving Good Drivers



- Evolving to drive fast without crashin (off road, obstacles)
- An interesting challenge of its own [77]
- Discovers optimal driving strategies (e.g. how to take curves)
- Works from range-finder & radar inpu
- Works from raw visual inputs (20×14 grayscale)

Evolving Warnings



- Evolving to estimate probability of crash
- Predicts based on subtle cues (e.g. skidding off the road)
- Compensates for disabled drivers
- Human drivers learn to drive with it!
- DEMO

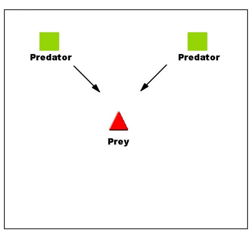
Transferring to the Physical World?

- Applied AI Gaia moving in an office environment
  - Sick laserfinder; Bumblebee digital camera
  - Driven by hand to collect data
- Learns collision warning in both cases
- Transfer to real cars?
- DEMO

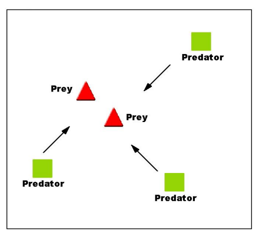
Applications to Artificial Life

- Gaining insight into neural structure
  - E.g. evolving a command neuron [2;35;61]
- Emergence of behaviors
  - Signaling, herding, hunting... [84;85;92]
- Future challenges
  - Emergence of language
  - Emergence of community behavior

Emergence of Cooperation and Competitio



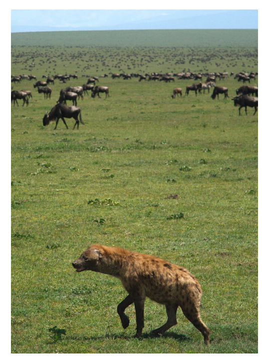
Predator cooperation



Predator, prey cooperation

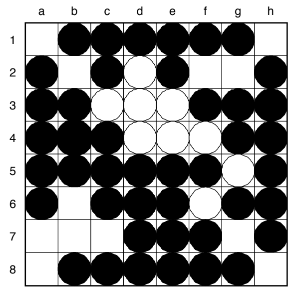
- Predator-prey simulations
  - Predator species, prey species
  - Prior work single pred/prey, team of pred/prey
- Simultaneous competitive and cooperative coevolution
- Understanding e.g. hyenas and zebras
  - Collaboration with biologists (Kay Holekamp, MSU)
- DEMO

Open Questions



- Role of communication
  - Stigmergy vs. direct communication in hunti
  - Quorum sensing in e.g. confronting lions
- Role of rankings
  - Efficient selection when evaluation is costly
- Role of individual vs. team rewards
- Can lead to general computational insights

Applications to Games



- Good research platform[45]
  - Controlled domains, clear performance, safe
  - Economically important; training games possible
- Board games: beyond limits of search
  - Evaluation functions in checkers, chess[9;19;20]
  - Filtering information in go, othello[48;74]
  - Opponent modeling in poker[42]

Video Games

- Economically and socially important
- GOFAI does not work well
  - Embedded, real-time, noisy, multiagent, changing
  - Adaptation a major component
- Possibly research catalyst for CI
  - Like board games were for GOFAI in the 1980s

Video Games (2)

- Can be used to build “mods” to existing games
  - Adapting characters, assistants, tools
- Can also be used to build new games
  - New genre: Machine Learning game

BotPrize Competition

- Turing Test for game bots: $10,000 prize
- Three players in Unreal Tournament:
  - Human confederate: tries to win
  - Software bot: pretends to be human
  - Human judge: tries to tell them apart!
- DEMO

Killer App: Evolving an Unreal Bot

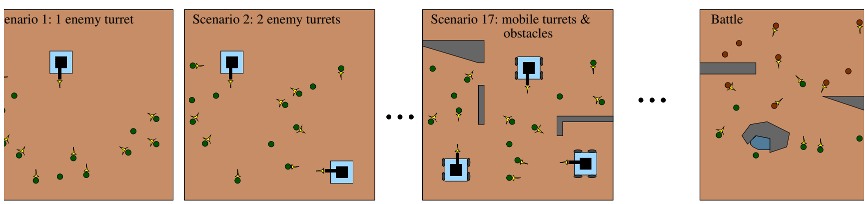
- Evolve basic strategies
  - Battle, chase, get-unstuck...
  - Can be extended to many other low-level behaviors
- Best bots judged 25-30% human
  - Vs. humans at 35-80%
- Fascinating challenges remain:
  - Judges can still differentiate in seconds
  - Judges lay cognitive, high-level traps

A New Genre: Machine Learning Games

NERO
NEURO EVOLVING ROBOTIC OPERATIVES

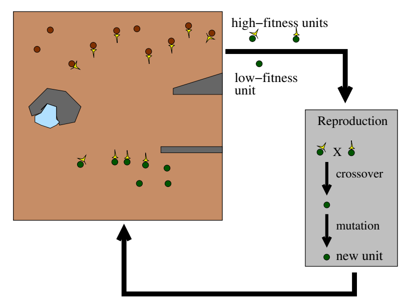
- E.g. NERO
  - Goal: to show that machine learning games are viable
  - Professionally produced by *Digital Media Collaboratory*, UT Austin
  - Developed mostly by volunteer undergraduates

NERO Gameplay



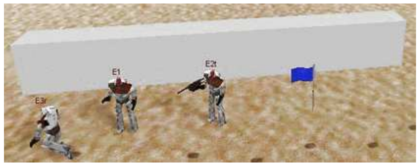
- Teams of agents trained to battle each other
  - Player trains agents through excercises
  - Agents evolve in real time
  - Agents and player collaborate in battle
- New genre: Learning *is* the game[30;70]
  - Challenging platform for reinforcement learning
  - Real time, open ended, requires discovery
- Try it out:
  - Available for download at `http://nerogame.org`
  - Open source research platform version at `opennero.googlecode.com`

Real-time NEAT



- A parallel, continuous version of NEAT[70]
- Individuals created and replaced every n ticks
- Parents selected probabilistically, weighted by fitness
- Long-term evolution equivalent to generational NEAT

NERO Player Actions



- Player can place items on the field

  e.g. static enemies, turrets, walls, rovers, flags

- Sliders specify relative importance of goals

  e.g. approach/avoid enemy, cluster/disperse, hit target, avoid fire...

- Networks evolved to control the agents
- DEMO

Numerous Other Applications

- Creating art, music, dance...[10;15;32;65]
- Theorem proving[14]
- Time-series prediction[43]
- Computer system optimization[24]
- Manufacturing optimization[28]
- Process control optimization[82;83]
- Measuring top quark mass[88]
- Etc.

Evaluation of Applications

- Neuroevolution strengths
  - Can work very fast, even in real-time
  - Potential for arms race, discovery
  - Effective in continuous, non-Markov domains
- Requires many evaluations
  - Requires an interactive domain for feedback
  - Best when parallel evaluations possible
  - Works with a simulator & transfer to domain

Conclusion

- NE is a powerful technology for sequential decision tasks
  - Evolutionary computation and neural nets are a good match
  - Lends itself to many extensions
  - Powerful in applications
- Easy to adapt to applications
  - Control, robotics, optimization
  - Artificial life, biology
  - Gaming: entertainment, training
- Lots of future work opportunities
  - Theory not well developed
  - Indirect encodings
  - Learning and evolution
  - Knowledge and interaction

References

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

[2] R. Aharonov-Barki, T. Beker, and E. Ruppin, Emergence of memory-Driven command neurons in evolved artificial agents, *Neural Computation*, 13(3):691–716 (2001).

[3] P. J. Angeline, G. M. Saunders, and J. B. Pollack, An evolutionary algorithm that constructs recurrent neural networks, *IEEE Transactions on Neural Networks*, 5:54–65 (1994).

[4] J. M. Baldwin, A new factor in evolution, *The American Naturalist*, 30:441–451, 536–553 (1896).

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

[6] B. D. Bryant and R. Miikkulainen, Neuroevolution for adaptive teams, in: *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, volume 3, 2194–2201, IEEE, Piscataway, NJ (2003).

[7] B. D. Bryant and R. Miikkulainen, Acquiring visibly intelligent behavior with example-guided neuroevolution, in: *Proceedings of the Twenty-Second National Conference on Artificial Intelligence*, 801–808, AAAI Press, Menlo Park, CA (2007).

[8] D. J. Chalmers, The evolution of learning: An experiment in genetic connectionism, in: *Connectionist Models: Proceedings of the 1990 Summer School*, D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, eds., 81–90, San Francisco: Morgan Kaufmann (1990).

[9] K. Chellapilla and D. B. Fogel, Evolution, neural networks, games, and intelligence, *Proceedings of the IEEE*, 87:1471–1496 (1999).

[10] C.-C. Chen and R. Miikkulainen, Creating melodies with evolving recurrent neural networks, in: *Proceedings of the INNS-IEEE International Joint Conference on Neural Networks*, 2241–2246, IEEE, Piscataway, NJ (2001).

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

[12] D. B. D'Ambrosio and K. O. Stanley, A novel generative encoding for exploiting neural network sensor and output geometry, in: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*, 974–981, ACM, New York, NY, USA (2007).

[13] D. B. D'Ambrosio and K. O. Stanley, Generative encoding for multiagent learning, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2008).

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

[15] G. Dubbin and K. O. Stanley, Learning to dance through interactive evolution, in: *Proceedings of the Eighth European Event on Evolutionary and Biologically Inspired Music, Sound, Art and Design*, Springer, Berlin (2010).

[16] J. Fan, R. Lau, and R. Miikkulainen, Utilizing domain knowledge in neuroevolution, in: *Machine Learning: Proceedings of the 20th Annual Conference* (2003).

[17] D. Floreano, P. Dürr, and C. Mattiussi, Neuroevolution: From architectures to learning, *Evolutionary Intelligence*, 1:47–62 (2008).

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

[19] D. B. Fogel, *Blondie24: Playing at the Edge of AI*, Morgan Kaufmann, San Francisco (2001).

[20] D. B. Fogel, T. J. Hays, S. L. Hahn, and J. Quon, Further evolution of a self-learning chess program, in: *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, IEEE, Piscataway, NJ (2005).

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

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

[23] F. Gomez, *Robust Non-Linear Control Through Neuroevolution*, Ph.D. thesis, Department of Computer Sciences, The University of Texas at Austin (2003).

[24] F. Gomez, D. Burger, and R. Miikkulainen, 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,

IEEE, Piscataway, NJ (2001).

[25] F. Gomez and R. Miikkulainen, Incremental evolution of complex general behavior, *Adaptive Behavior*, 5:317–342 (1997).

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

[27] F. Gomez, J. Schmidhuber, and R. Miikkulainen, Accelerated neural evolution through cooperatively coevolved synapses, *Journal of Machine Learning Research*, 9:937–965 (2008).

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

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

[30] E. J. Hastings, R. K. Guha, and K. O. Stanley, Automatic content generation in the galactic arms race video game, *IEEE Transactions on Computational Intelligence and AI in Games*, 1:245–263 (2009).

[31] G. E. Hinton and S. J. Nowlan, How learning can guide evolution, *Complex Systems*, 1:495–502 (1987).

[32] A. K. Hoover, M. P. Rosario, and K. O. Stanley, Scaffolding for interactively evolving novel drum tracks for existing songs, in: *Proceedings of the Sixth European Workshop on Evolutionary and Biologically Inspired Music, Sound, Art and Design*, Springer, Berlin (2008).

[33] G. S. Hornby, S. Takamura, J. Yokono, O. Hanagata, M. Fujita, and J. Pollack, 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, Springer, Berlin (2000).

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

[35] A. Keinan, B. Sandbank, C. C. Hilgetag, I. Meilijson, and E. Ruppin, Axiomatic scalable neurocontroller analysis via the Shapley value, *Artificial Life*, 12:333–352 (2006).

[36] N. Kohl and R. Miikkulainen, Evolving neural networks for strategic decision-making problems, *Neural Networks*,

22:326–337 (2009).

[37] N. Kohl, K. O. Stanley, R. Miikkulainen, M. Samples, and R. Sherony, Evolving a real-world vehicle warning system, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2006).

[38] J. Lehman and K. O. Stanley, Efficiently evolving programs through the search for novelty, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2010).

[39] J. Lehman and K. O. Stanley, Revising the evolutionary computation abstraction: Minimal criteria novelty search, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2010).

[40] J. Lehman and K. O. Stanley, Abandoning objectives: Evolution through the search for novelty alone, *Evolutionary Computation* (in press).

[41] Y. Liu, X. Yao, and T. Higuchi, Evolutionary ensembles with negative correlation learning, *IEEE Transactions on Evolutionary Computation*, 4:380–387 (2000).

[42] A. J. Lockett, C. L. Chen, and R. Miikkulainen, Evolving explicit opponent models in game playing, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2007).

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

[44] P. McQuesten, *Cultural Enhancement of Neuroevolution*, Ph.D. thesis, Department of Computer Sciences, The University of Texas at Austin, Austin, TX (2002). Technical Report AI-02-295.

[45] R. Miikkulainen, B. D. Bryant, R. Cornelius, I. V. Karpov, K. O. Stanley, and C. H. Yong, Computational intelligence in games, in: *Computational Intelligence: Principles and Practice*, G. Y. Yen and D. B. Fogel, eds., IEEE Computational Intelligence Society, Piscataway, NJ (2006).

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

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

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

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

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

[51] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette, Evolutionary algorithms for reinforcement learning, *Journal of Artificial Intelligence Research*, 11:199–229 (1999).

[52] S. Nolfi, J. L. Elman, and D. Parisi, Learning and evolution in neural networks, *Adaptive Behavior*, 2:5–28 (1994).

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

[54] S. Nolfi and D. Parisi, Good teaching inputs do not correspond to desired responses in ecological neural networks, *Neural Processing Letters*, 1(2):1–4 (1994).

[55] D. Pardoe, M. Ryoo, and R. Miikkulainen, Evolving neural network ensembles for control problems, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2005).

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

[57] J. Reisinger and R. Miikkulainen, Acquiring evolvability through adaptive representations, in: *Proceeedings of the Genetic and Evolutionary Computation Conference*, 1045–1052 (2007).

[58] J. Reisinger, K. O. Stanley, and R. Miikkulainen, Evolving reusable neural modules, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, 69–81 (2004).

[59] C. D. Rosin and R. K. Belew, New methods for competitive coevolution, *Evolutionary Computation*, 5:1–29 (1997).

[60] T. P. Runarsson and M. T. Jonsson, Evolution and design of distributed learning rules, in: *Proceedings of The First IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, 59–63, IEEE, Piscataway, NJ (2000).

[61] E. Ruppin, Evolutionary autonomous agents: A neuroscience perspective, *Nature Reviews Neuroscience* (2002).

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

[63] J. Schrum and R. Miikkulainen, Evolving multi-modal behavior in npcs, in: *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, IEEE, Piscataway, NJ (2009).

[64] J. Schrum and R. Miikkulainen, Evolving agent behavior in multiobjective domains using fitness-based shaping, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2010).

[65] J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, and K. O. Stanley, Picbreeder: Evolving pictures collaboratively online, in: *Proceedings of Computer Human Interaction Conference*, ACM, New York (2008).

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

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

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

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

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

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

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

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

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

[75] D. G. Stork, S. Walker, M. Burns, and B. Jackson, Preadaptation in neural circuits, in: *International Joint Conference on Neural Networks* (Washington, DC), 202–205, IEEE, Piscataway, NJ (1990).

[76] M. Taylor, S. Whiteson, and P. Stone, Comparing evolutionary and temporal difference methods in a reinforcement learning domain, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2006).

[77] J. Togelius and S. M. Lucas, Evolving robust and specialized car racing skills, in: *IEEE Congress on Evolutionary Computation*, 1187–1194, IEEE, Piscataway, NJ (2006).

[78] J. Urzelai, D. Floreano, M. Dorigo, and M. Colombetti, Incremental robot shaping, *Connection Science*, 10:341–360 (1998).

[79] V. Valsalam and R. Miikkulainen, Evolving symmetric and modular neural networks for distributed control, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, 731–738 (2009).

[80] V. Valsalam and R. Miikkulainen, Evolving symmetry for modular system design, *IEEE Transactions on Evolutionary Computation* (in press).

[81] V. K. Valsalam and R. Miikkulainen, Modular neuroevolution for multilegged locomotion, in: *Proceedings of the Genetic and Evolutionary Computation Conference GECCO 2008*, 265–272, ACM, New York, NY, USA (2008).

[82] A. van Eck Conradie, R. Miikkulainen, and C. Aldrich, Adaptive control utilising neural swarming, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, W. B. Langdon, E. Cantú-Paz, K. E. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. K. Burke, and N. Jonoska, eds., 60–67, San Francisco: Morgan Kaufmann (2002).

[83] A. van Eck Conradie, R. Miikkulainen, and C. Aldrich, Intelligent process control utilizing symbiotic memetic neuroevolution, in: *Proceedings of the 2002 Congress on Evolutionary Computation*, 623–628 (2002).

[84] G. M. Werner and M. G. Dyer, Evolution of communication in artificial organisms, in: *Proceedings of the Workshop on Artificial Life (ALIFE '90)*, C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, eds., 659–687, Reading, MA: Addison-Wesley (1992).

[85] G. M. Werner and M. G. Dyer, Evolution of herding behavior in artificial animals, in: *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, J.-A. Meyer, H. L. Roitblat, and S. W. Wilson, eds., Cambridge, MA: MIT Press (1992).

[86] S. Whiteson, N. Kohl, R. Miikkulainen, and P. Stone, Evolving keepaway soccer players through task decomposition, *Machine Learning*, 59:5–30 (2005).

[87] S. Whiteson and P. Stone, Evolutionary function approximation for reinforcement learning, *Journal of Machine Learning Research*, 7:877–917 (2006).

[88] S. Whiteson and D. Whiteson, Stochastic optimization for collision selection in high energy physics, in: *Proceedings of the Nineteenth Annual Innovative Applications of Artificial Intelligence Conference* (2007).

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

[90] A. P. Wieland, Evolving controls for unstable systems, in: *Connectionist Models: Proceedings of the 1990 Summer School*, D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, eds., 91–102, San Francisco: Morgan Kaufmann (1990).

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

[92] C. H. Yong and R. Miikkulainen, Cooperative coevolution of multi-agent systems, Technical Report AI07-338, Department of Computer Sciences, The University of Texas at Austin (2007).

[93] C. H. Yong, K. O. Stanley, R. Miikkulainen, and I. Karpov, Incorporating advice into neuroevolution of adaptive agents, in: *Proceedings of the Second Artificial Intelligence and Interactive Digital Entertainment Conference*, 98–104, AAAI Press, Menlo Park, CA (2006).