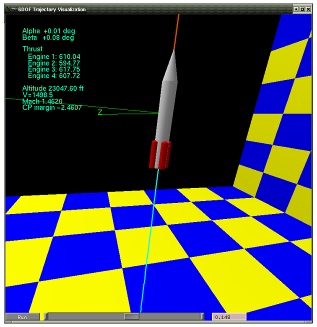
Evolving Neural Networks

Risto Miikkulainen

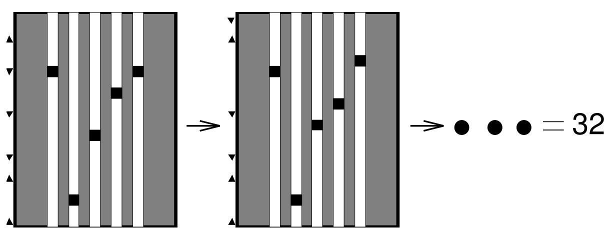
Department of Computer Science
The University of Texas at Austin
risto@cs.utexas.edu

Why Neuroevolution?



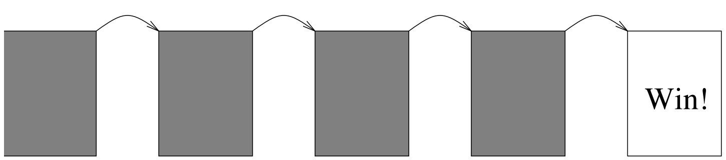
- Neural nets powerful in many statistical domains
  - E.g. control, pattern recognition, prediction, decision making
  - Where no good theory of the domain exists
- Good supervised training algorithms exist
  - Learn a nonlinear function that matches the examples
- What if correct outputs are not known?

Sequential Decision Tasks



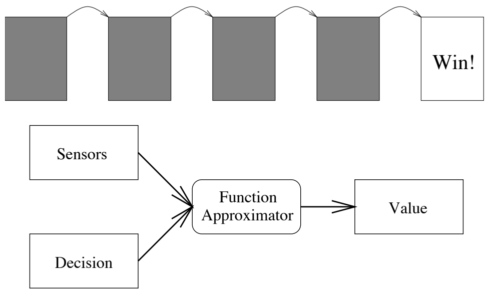
- Sequence of decisions creates a sequence of states
- No targets: Performance evaluated after several decisions
- Many important real-world domains:
  - Robot/vehicle/traffic control
  - Computer/manufacturing/process optimization
  - Game playing

Forming Decision Strategies



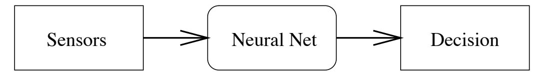
- Traditionally designed by hand
  - Too complex: Hard to anticipate all scenarios
  - Too inflexible: Cannot adapt on-line
- Need to discover through exploration
  - Based on sparse reinforcement
  - Associate actions with outcomes

Standard Reinforcement Learning



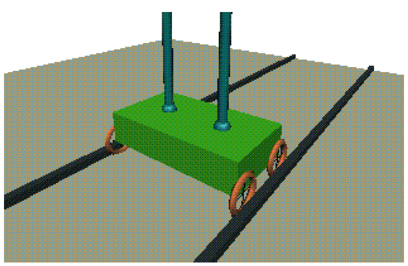
- AHC, Q-learning, Temporal Differences
  - Generate targets through prediction errors
  - Learn when successive predictions differ
- Predictions represented as a value function
  - Values of alternatives at each state
- Difficult with large/continuous state and action spaces
- Difficult with hidden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

Neuroevolution (NE) Reinforcement Learning



- 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 [103]
  - Deep Reinforcement Learning [82]

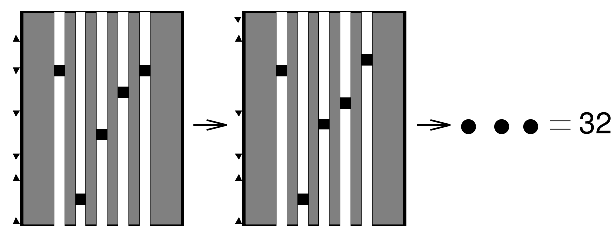
How Well Does It Work?



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 2-3 orders of magnitude faster than standard RL [30]
- NE can solve harder problems

Role of Neuroevolution

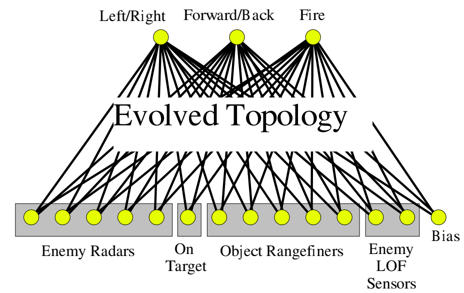


- Powerful method for sequential decision tasks [18,30,65,121]
  - Optimizing existing tasks
  - Discovering novel solutions
  - Making new applications possible
- Also may be useful in supervised tasks [60,72]
  - Especially when network topology important
- A unique model of biological adaptation/development [67,80,115]

Outline

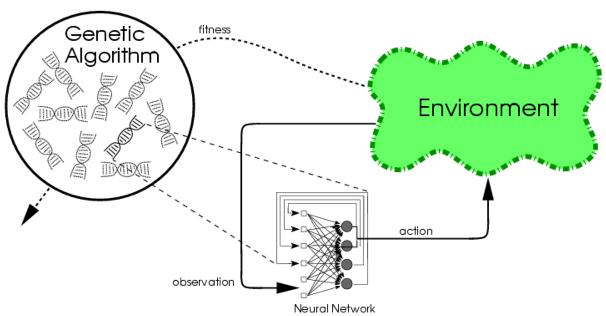
- Basic neuroevolution techniques
- Advanced techniques
  - E.g. combining learning and evolution; novelty search
- Extensions to applications
- Application examples
  - Control, Robotics, Artificial Life, Games

Neuroevolution Decision Strategies



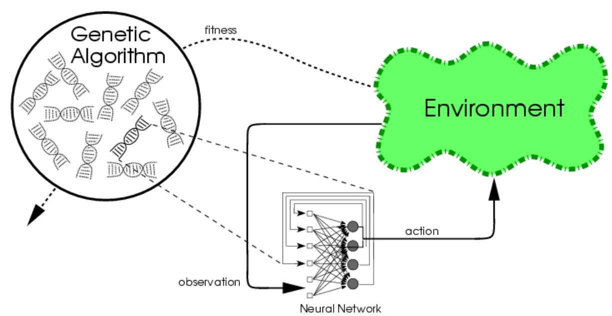
- Input variables describe the state observed through sensors
- Output variables describe actions
- Network between input and output:
  - Nonlinear hidden nodes
  - Weighted connections
- Execution:
  - Numerical activation of input
  - Performs a nonlinear mapping
  - Memory in recurrent connections (POMDP!)

Conventional Neuroevolution (CNE) I



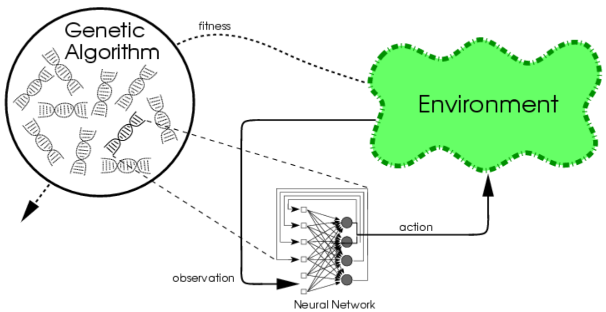
- Evolving connection weights in a population of networks [60,81,121,122]
- Chromosomes are strings of connection weights (bits or real)
  - E.g. 10010110101100101111001
  - Usually fully connected, fixed topology
  - Initially random

Conventional Neuroevolution II



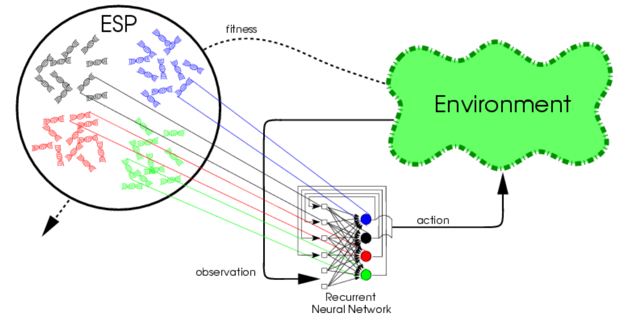
- Parallel search for a solution network
  - Each NN evaluated in the task
  - Good NN reproduce through crossover, mutation
  - Bad thrown away
- Natural mapping between genotype and phenotype
  - GA and NN are a good match!

Problems with CNE



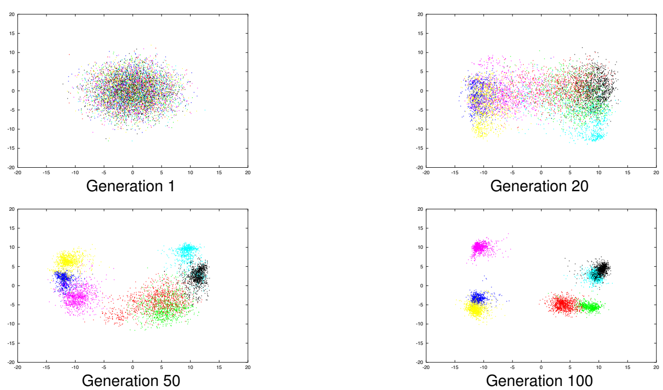
- Evolution converges the population (as usual with EAs)
  - Diversity is lost; progress stagnates
- Competing conventions
  - Different, incompatible encodings for the same solution
- Too many parameters to be optimized simultaneously
  - Thousands of weight values at once

Advanced NE 1: Evolving Partial Networks I



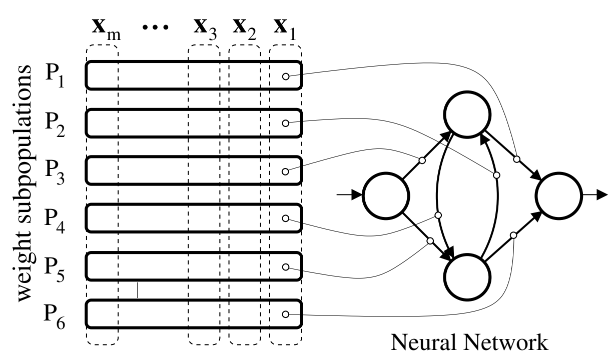
- Evolving individual neurons to cooperate in networks [1,64,72]
- E.g. Enforced Sub-Populations (ESP [25])
  - Each (hidden) neuron in a separate subpopulation
  - Fully connected; weights of each neuron evolved
  - Populations learn compatible subtasks

Evolving Neurons with ESP



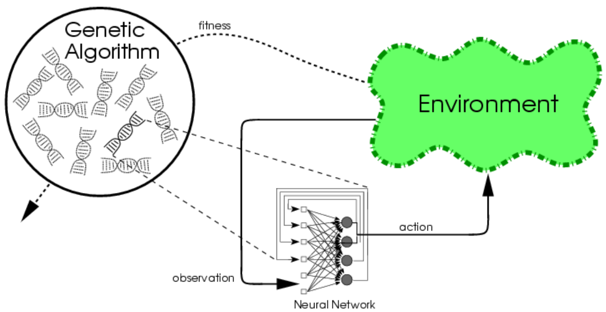
- Evolution encourages diversity automatically
  - Good networks require different kinds of neurons
- Evolution discourages competing conventions
  - Neurons optimized for compatible roles
- Large search space divided into subtasks
  - Optimize compatible neurons

Evolving Partial Networks II



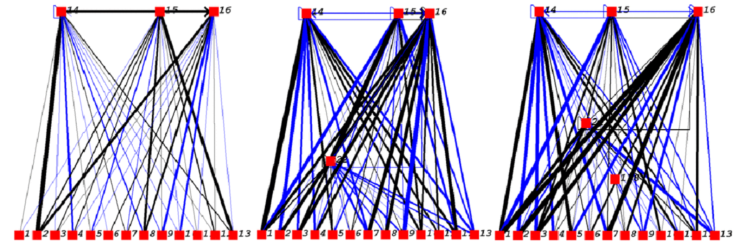
- Extend the idea to evolving connection weights
- E.g. Cooperative Synapse NeuroEvolution (CoSyNE [30])
  - Connection weights in separate subpopulations
  - Networks formed by combining neurons with the same index
  - Networks mutated and recombined; indices permutated
- Sustains diversity, results in efficient search

Advanced NE 2: Evolutionary Strategies



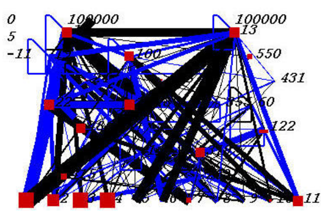
- Evolving complete networks with ES (CMA-ES[37])
- Small populations, no crossover
- Instead, intelligent mutations
  - Adapt covariance matrix of mutation distribution
  - Take into account correlations between weights
- Smaller space, less convergence, fewer conventions

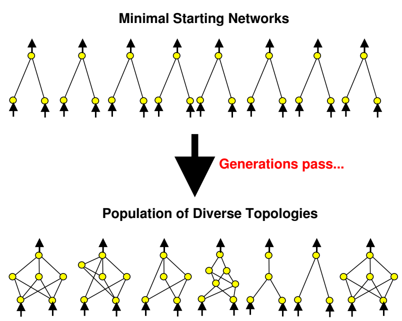
Advanced NE 3: Evolving Network Structure



- Optimizing connection weights and network topology[3,18,23,123]
- E.g. Neuroevolution of Augmenting Topologies (NEAT[94,97])
- Based on *Complexification*
- Of networks:
  - Mutations to add nodes and connections
- Of behavior:
  - Elaborates on earlier behaviors

Why Complexification?

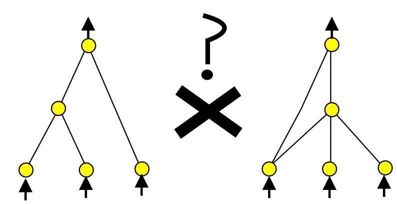




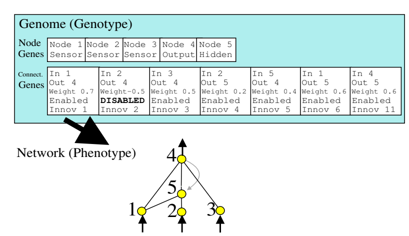
- Challenge with NE: Search space is very large
- Complexification keeps the search tractable
  - Start simple, add more sophistication
- Incremental construction of intelligent agents

How Can Crossover be Implemented?

- Problem: Structures do not match

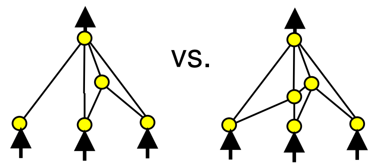


- Solution: Utilize historical markings



How Can Innovation Survive?

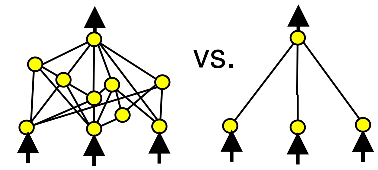
- Problem: Innovations have initially low fitness



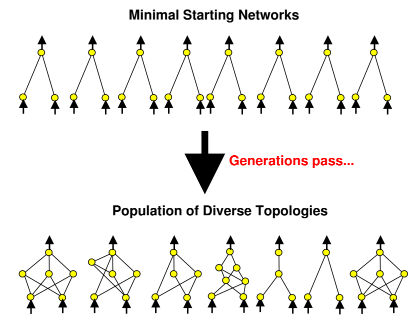
- Solution: Speciate the population
  - Innovations have time to optimize
  - Mitigates competing conventions
  - Promotes diversity

How Can We Search in Large Spaces?

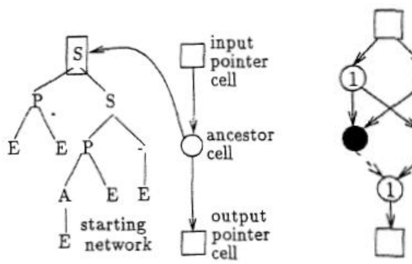
- Need to optimize not just weights but also topologies



- Solution: Start with minimal structure and complexify
  - Hidden nodes, connections, input features[119]

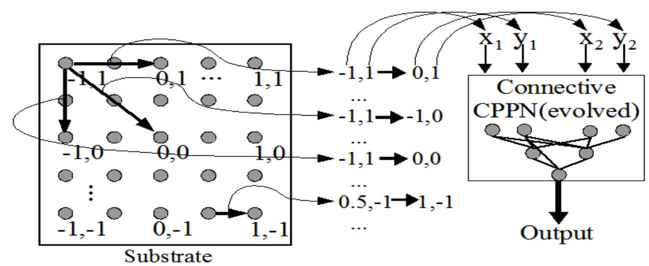


Advanced NE 4: Indirect Encodings I

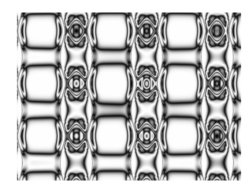


- Instructions for constructing the network evolved
  - Instead of specifying each unit and connection[3,18,58,90,123]
- E.g. Cellular Encoding (CE[32])
- Grammar tree describes construction
  - Sequential and parallel cell division
  - Changing thresholds, weights
  - A "developmental" process that results in a network

Indirect Encodings II

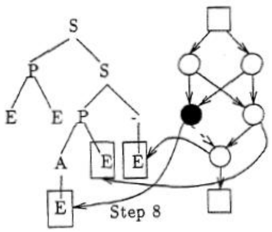


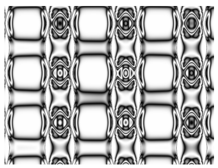
- Encode the networks as spatial patterns
- E.g. Hypercube-based NEAT (HyperNEAT[14])



- Evolve a neural network (CPPN) to generate spatial patterns
  - 2D CPPN: (x, y) input $\rightarrow$ grayscale output
  - 4D CPPN: (x_1, y_1, x_2, y_2) input $\rightarrow$ w output
  - Connectivity and weights can be evolved indirectly
  - Works with very large networks (millions of connections)

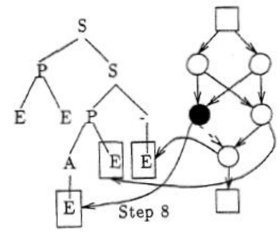
Properties of Indirect Encodings I

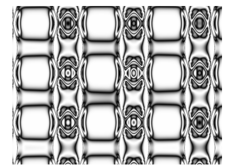




- Smaller search space
- Avoids competing conventions
- Describes classes of networks efficiently
- Modularity, reuse of structures
  - Recurrency symbol in CE: XOR → parity
  - Repetition with variation in CPPNs
  - Useful for evolving morphology

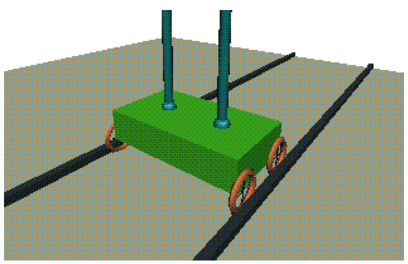
Properties of Indirect Encodings II





- Not fully explored (yet)
  - See e.g. GDS track at GECCO
- Promising current work
  - More general L-systems; developmental codings; embryogeny[98]
  - Scaling up spatial coding[15,24]
  - Genetic Regulatory Networks[76]
  - Evolution of symmetries[112]

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

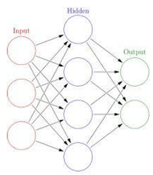
Two poles, no velocities, damping fitness[30]

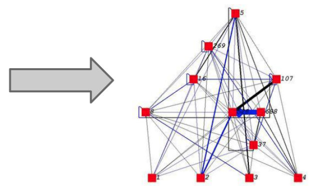
- Advanced methods better than CNE
- Advanced methods still under development
- Indirect encodings future work

Further NE Techniques

- Incremental and multiobjective evolution[27,84,107,122]
- Utilizing population culture[6,56,102]
- Utilizing evaluation history[52]
- Evolving NN ensembles and modules[38,51,71,77,117]
- Evolving transfer functions and learning rules[9,79,101]
- Bilevel optimization of NE[50]
- Combining learning and evolution
- Evolving for novelty

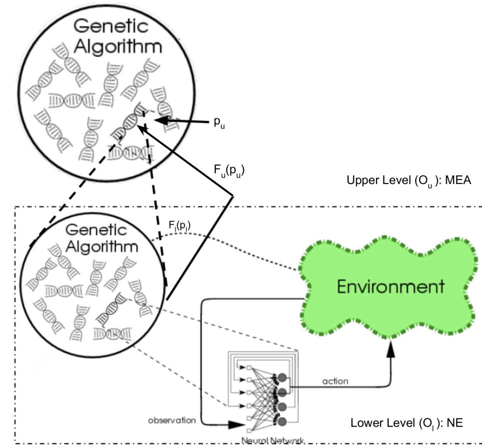
Bilevel Optimization of NE





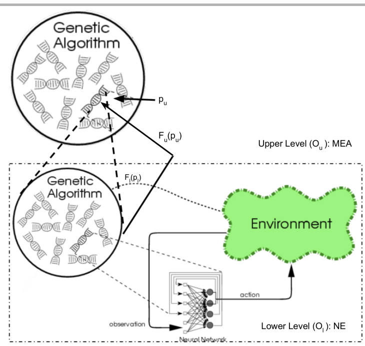
- Good performance depends on good parameters
  - Population size, mutation rate & amount, crossover rate...
- Neuroevolution methods becoming more complex
  - More parameters, harder to optimize by hand

Solution: Bilevel Evolution



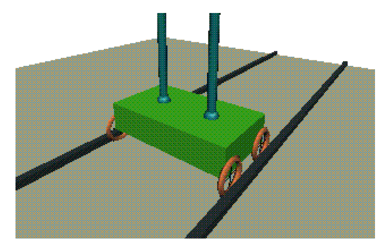
- Two nested evolutionary processes:
  - Lower level evolution to optimize the NN
  - Upper level evolution to optimize the parameters
- Intractable to run hierarchically

Solution: MEA



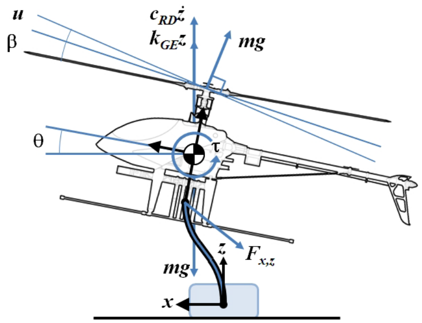
- Approximate fitness with a regression model[50,92]
  - Only evaluate individuals with predicted high fitness
  - Random forest regression (robust to noise)
- Use metrics, not parameters, to fit regression
  - E.g. Best mean fitness, diversity, fitness growth & stdev...
- Run for limited time only
  - Fast learning correlates with performance
- →MEA will scale up

MEA in Double Pole-Balancing



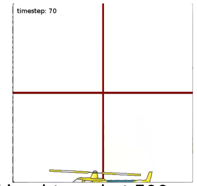
- Five parameters evolved
  - Mutation rate & amount, replacement rate, initial weight range, population size
  - →Optimal population size 17 (instead of 400)
- Fifteen parameters evolved
  - Selection, crossover types and probabilities
  - More than can be optimized by hand
  - →Significantly better performance
- Bilevel optimization makes better solutions possible!

MEA in Helicopter Hovering

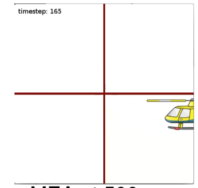


- A challenging benchmark
  - RL, NE solutions exist
- Eight parameters optimized from literature[43]
  - →Significantly better performance
  - →Optimal population size 32 (instead of 50)
  - →Fast learning predicts good performance

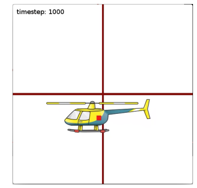
Helicopter Hovering Demo



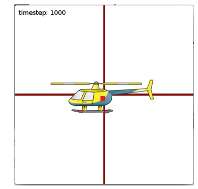
Hand-tuned at 500 gen



MEA at 500 gen



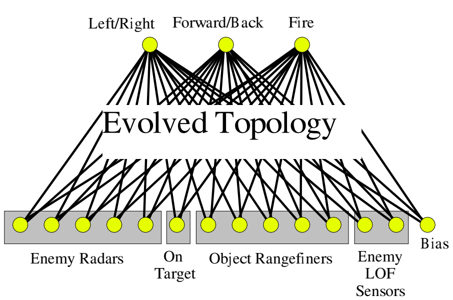
Hand-tuned at 5000 gen



MEA at 5000 gen

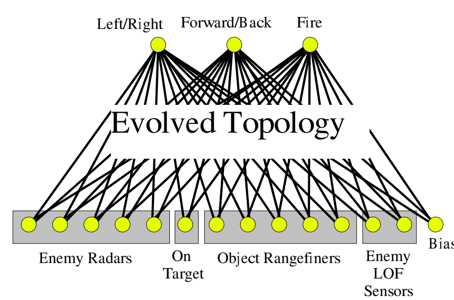
- Bilevel optimization → better performance
  - More parameters, more complex algorithms
- Future work: tune parameters dynamically
  - Based on state-of-evolution metrics
  - E.g. decrease mutation gradually

Combining Learning and Evolution



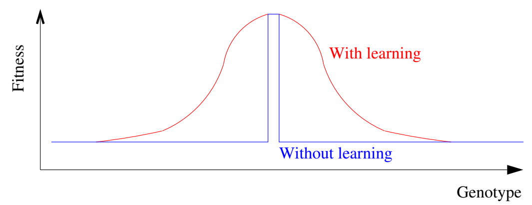
- Good learning algorithms exist for NN
  - Why not use them as well?
- Evolution provides structure and initial weights
- Fine tune the weights by learning

Lamarckian Evolution



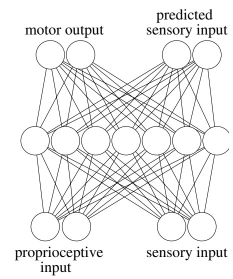
- Lamarckian evolution is possible[8,32]
  - Coding weight changes back to chromosome
- Difficult to make it work
  - Diversity reduced; progress stagnates

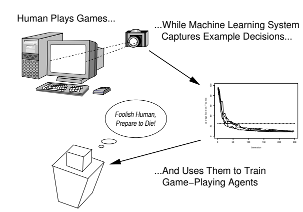
Baldwin Effect



- Learning can guide Darwinian evolution as well[5,32,34]
  - 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

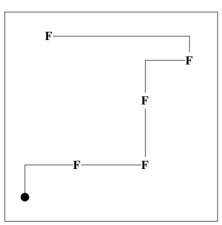
Where to Get Learning Targets?

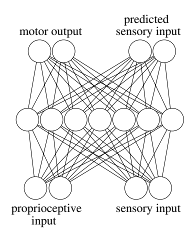




- From a related task[67]
  - Useful internal representations
- Evolve the targets[70]
  - Useful training situations
- From Q-learning equations[118]
  - When evolving a value function
- Utilize Hebbian learning[20,95,110]
  - Correlations of activity
- From the population[56,102]
  - Social learning
- From humans[8]
  - E.g. expert players, drivers

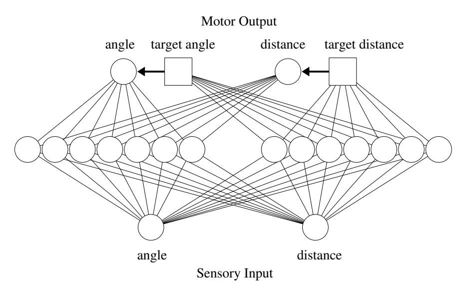
Targets from a Related Task





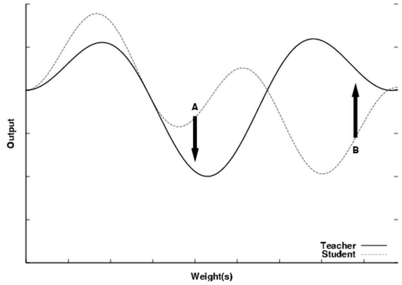
- Learning in a related task is sufficient
- E.g. foraging for food in a microworld[67]
  - Network sees the state, outputs motor commands
  - Trained with backprop to predict the next input
  - Training emphasizes hidden-layer representations
  - Allows more accurate evaluations

Evolving the Targets



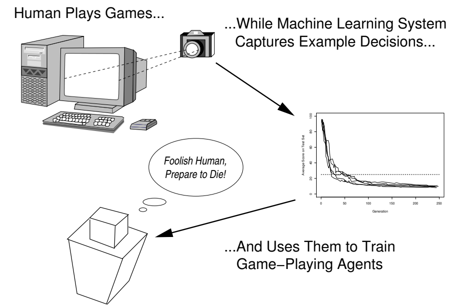
- Evolve extra outputs to provide targets
- E.g. in the foraging task[70]
  - Motor outputs and targets with separate hidden layers
  - Motor weights trained with backprop, targets evolved
  - Targets do not correspond to optimal performance: Direct system towards useful learning experiences

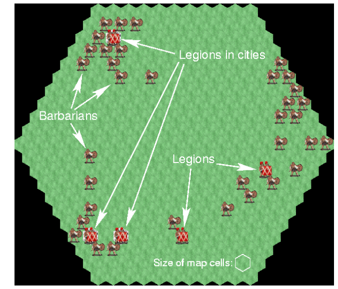
Targets from the Population



- Train new offspring to imitate parents/champion[56]
  - Trained in population "culture"
- Local search around good individuals
  - Limited training: 8-20 backprop iterations
- Becomes part of the evaluation
  - Individuals evolve to anticipate training
  - Perform poorly at birth, well after training
- Evolution discovers optimal starting points for learning!

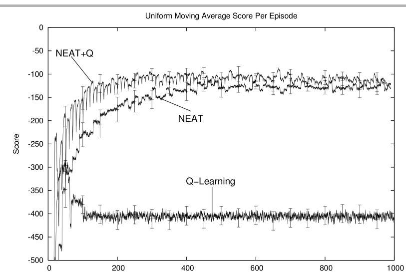
Targets from Humans





- Humans can demonstrate desired behavior
- E.g. fine tuning game agents[8]
  - Human observer identifies suboptimal behavior
  - Drives the NPC with a joystick
  - Agent placed in the same input situation
  - Backpropagate from human actions

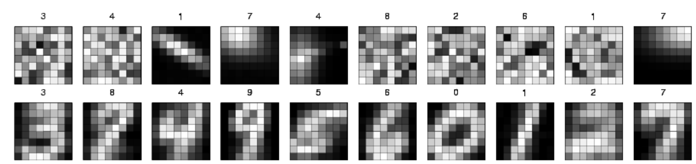
Targets from Q-Learning



(a) Mountain Car

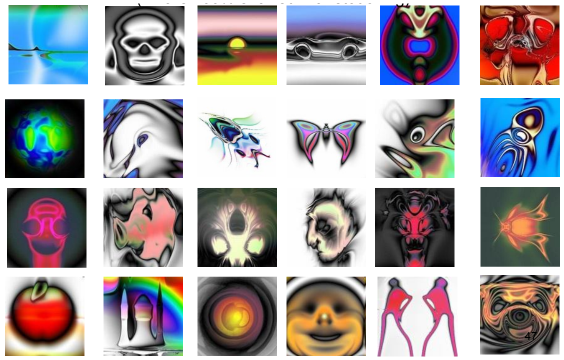
- E.g. NEAT+Q[118]: Evolve network to represent value function
  - Input is the state, outputs are Q-values of actions
- Form targets according to Q-learning equations
  - Compare successive Q-values, use backprop to train
- Improves evolution of a value function
  - Faster than NEAT alone, better than Q-learning
- Utilize both evolution and on-line learning

No Targets: Unsupervised Learning



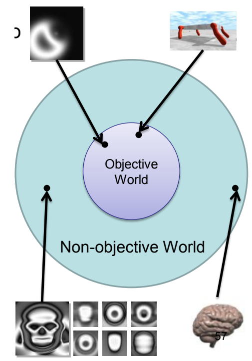
- Hebbian adaptation during performance[20,95]
- E.g. handwritten character recognition[110]
  - Evolution determines the starting point
  - Competitive learning finishes the design
- Starting points are poor recognizers
  - Only bias learning away from local minima
- Synergetic effect: Evolution utilizes learning
- Future work: Constructing developmental systems

Evolving Novelty



- Motivated by humans as fitness functions
- E.g. picbreeder.com, endlessforms.com[87]
  - CPPNs evolved; Human users select parents
- No specific goal
  - Interesting solutions preferred
  - Similar to biological evolution?

Novelty Search



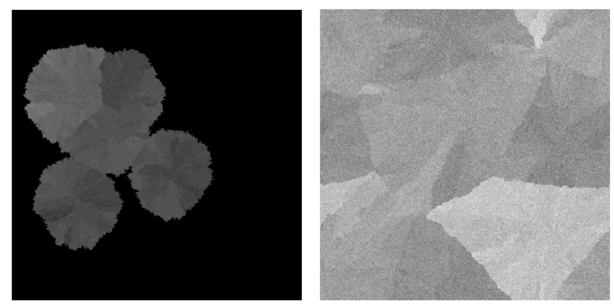
- Reward maximally different solutions
  - Can be a secondary, diversity objective[66]
  - Or, even as the only objective[44,47]
- To be different, need to capture structure
  - Problem solving as a side effect
- Potential for innovation
- Model of biological evolution, niching[46]
- Needs to be understood better

Novelty Search Mechanisms: Deception

gen 12 gen 20 gen 36 gen 49 gen 74

- Deception is not a problem
  - Stepping stones survive if they are novel
- Important e.g. in evolution of cognitive behavior
  - Memory, learning, communication are deceptive[45]
- Difficult to discover with objective-based search

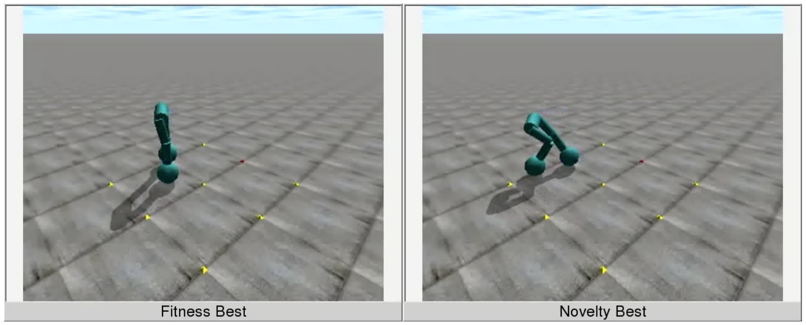
Novelty Search Mechanisms: Evolvability



(c) Extinction Random - 2,000 Gens. (d) Extinction Random - 15,000 Gens.

- Extinction events helpful[46]
  - E.g. 90% of population decimated occasionally
  - Remaining lineages radiate through niches
- They select for more evolvable lineages
  - Discover better solutions faster
- Harmful in objective-based search!

Novelty Search Demo



- Fitness-based evolution is rigid
  - Requires gradual progress
- Novelty-based evolution is more innovative, natural
  - Allows building on deceptive solutions
- DEMO

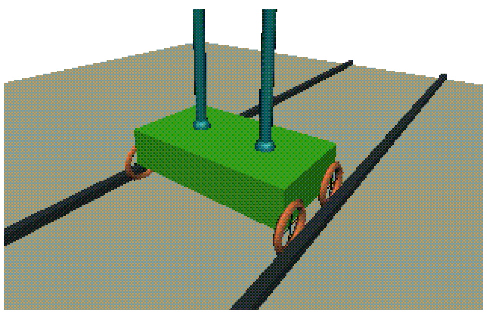
Extending NE to Applications

- Control
- Robotics
- Artificial life
- Gaming

Issues:

- Facilitating robust transfer from simulation[29,108]
- Utilizing problem symmetry and hierarchy[41,111,112]
- Utilizing coevolution[78,99]
- Evolving multimodal behavior[83,84,117]
- Evolving teams of agents[7,96,125]
- Making evolution run in real-time[96]

Applications to Control



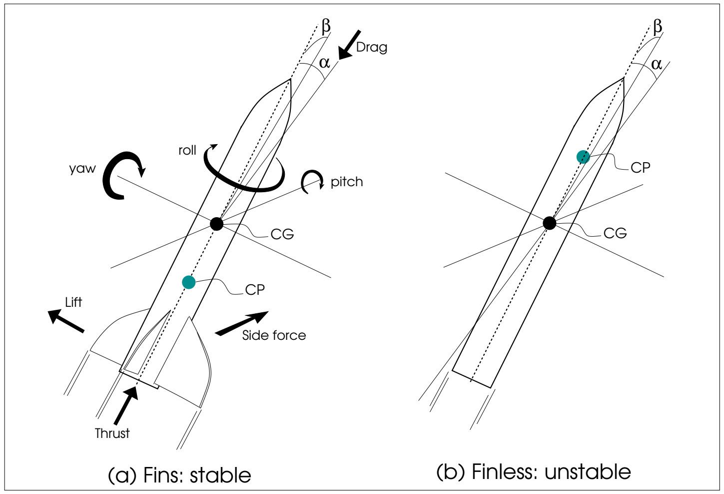
- Pole-balancing benchmark
  - Originates from the 1960s
  - Original 1-pole version too easy
  - Several extensions: acrobat, jointed, 2-pole, particle chasing[71]
  - DEMO
- Good surrogate for other control tasks
  - Vehicles and other physical devices
  - Process control[113]

Controlling a Finless Rocket

Task: Stabilize a finless version of the Interorbital Systems RSX-2 sounding rocket[28]

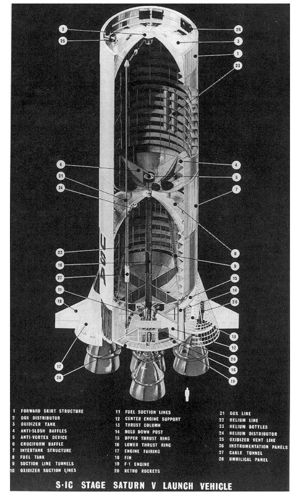
- Scientific measurements in the upper atmosphere
- 4 liquid-fueled engines with variable thrust
- Without fins will fly much higher for same amount of fuel

Rocket Stability



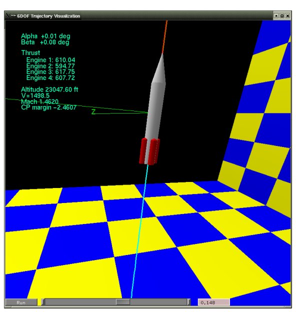
(a) Fins: stable (b) Finless: unstable

Active Rocket Guidance



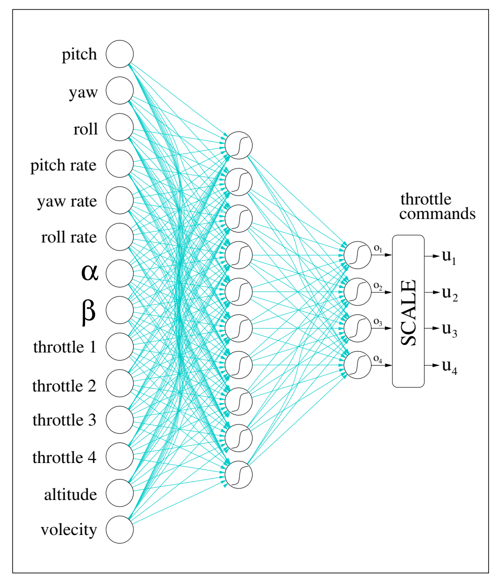
- Used on large scale launch vehicles (Saturn, Titan)
- Typically based on classical linear feedback control
- High level of domain knowledge required
- Expensive, heavy

Simulation Environment: JSBSim

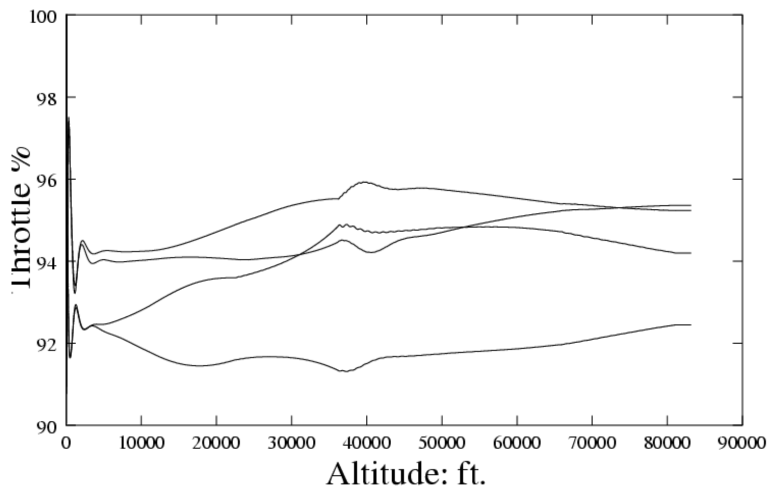


- General rocket simulator
- Models complex interaction between airframe, propulsion, aerodynamics, and atmosphere
- Used by IOS in testing their rocket designs
- Accurate geometric model of the RSX-2

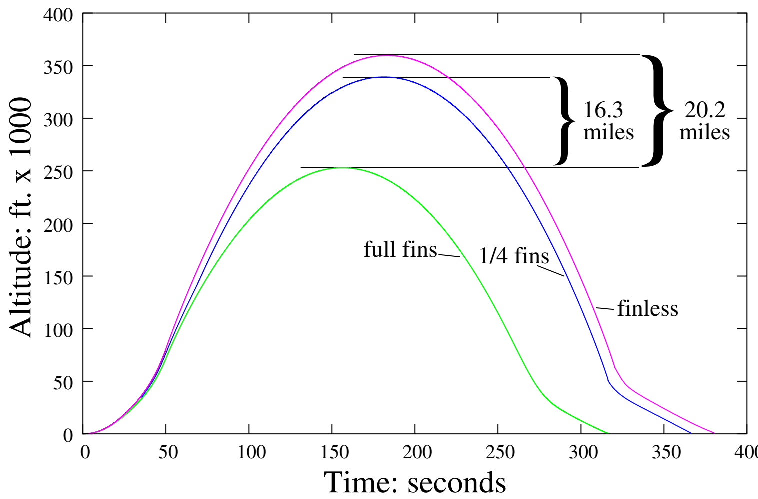
Rocket Guidance Network



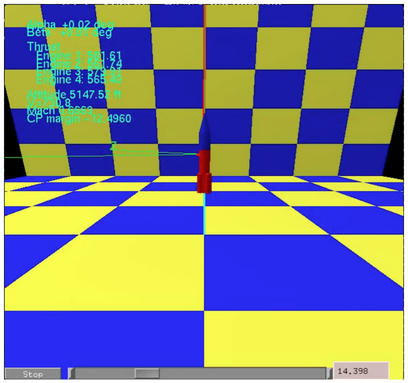
Results: Control Policy

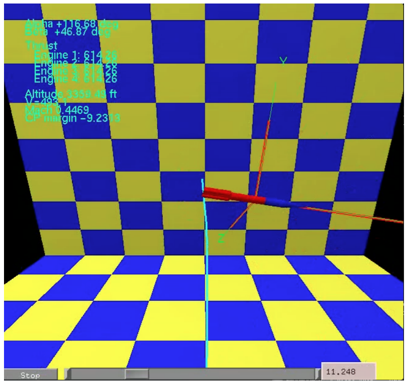


Results: Apogee



Finless Rocket Control Demo

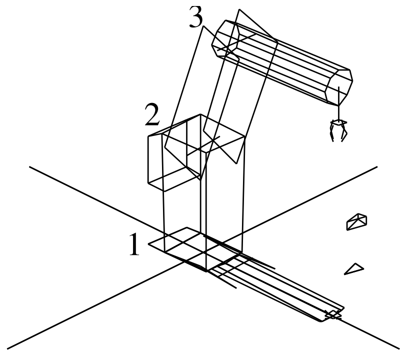

Evolved active stabilization


No active stabilization

Transfer to Physical RSX-2?

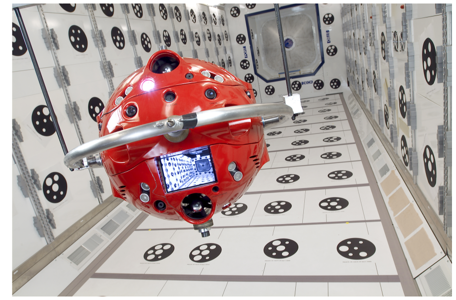
- More coarse control: two throttle levels (done)
- Wind disturbances (done)
  - Robust as is
  - Improve with trajectory noise
- Rid of α and β?
- Actual rocket launch?

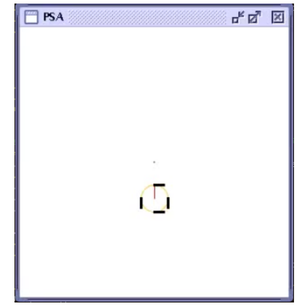
Applications to Robotics



- Controlling a robot arm[63]
  - Compensates for an inop motor
- Robot walking[36,89,111]
  - Various physical platforms
- Mobile robots[12,19,68,93]
  - Transfers from simulation to physical robots
  - Evolution possible on physical robots

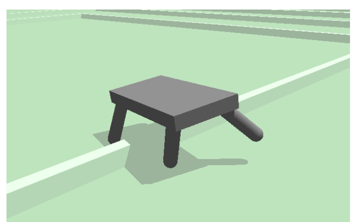
Personal Satellite Assistant





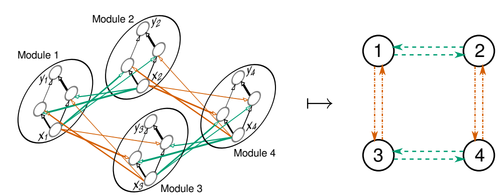
- Floating robot assistant to an astronaut
  - Needs to stay close but not crash
  - Limited thrusters: Difficult to control
- Novel control strategies can be evolved
  - Stop on a spot by making a circle![93]
- DEMO

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[109,111,112]
  - A neural network controls each leg
  - Connections between controllers evolved through symmetry breaking
  - Connections within individual controllers evolved through neuroevolution

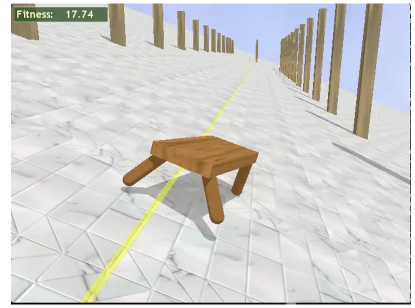
Versatile, Robust Gaits

Different gaits

Obstacle field

- Different gaits on flat ground
  - Pronk, pace, bound, trot
  - Changes gait to get over obstacles
- DEMO

Innovative, Effective Solutions

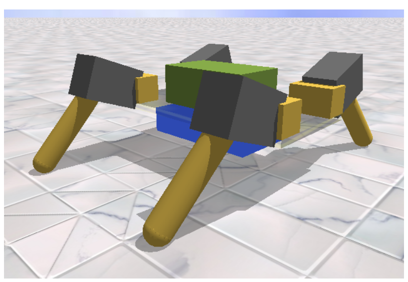


Evolved

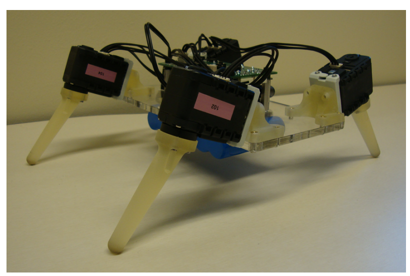
Handcoded

- Asymmetric gait on inclines
  - One leg pushes up, others forward
  - Hard to design by hand
- DEMO

Transfer to a Physical Robot I



Simulated

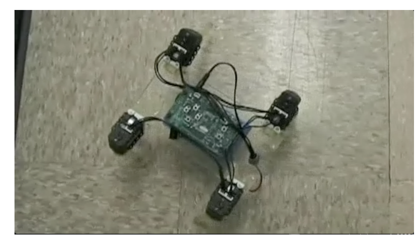


Real

- 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[108]
  - Noise to actuators during simulation
  - Generalizes to different surfaces, motor speeds
- DEMO

Transfer to a Physical Robot II

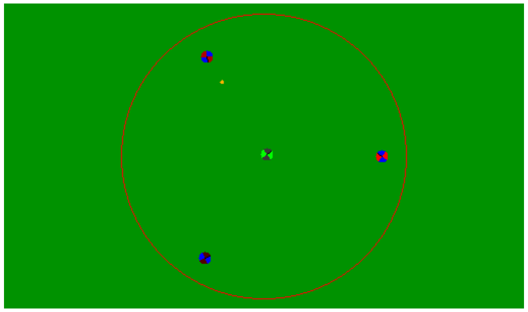
Evolved



Handcoded

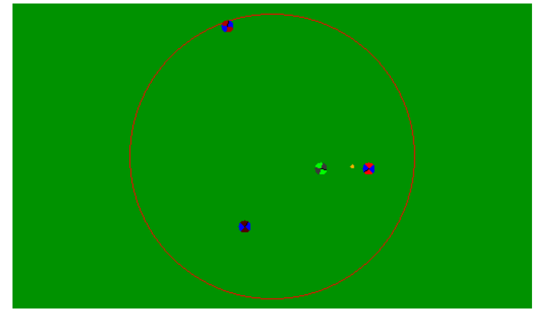
- Evolved a solution for three-legged walking!
- DEMO

Robotic Soccer



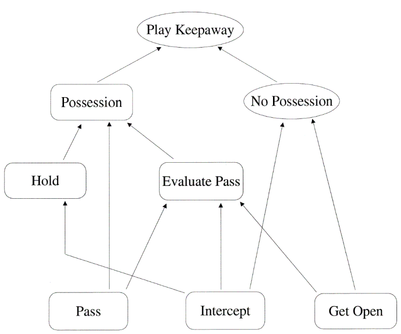
- E.g. robocup soccer “Keepaway” task[117]
- Three keepers, one (algorithmic) taker
- Includes many behaviors:
  Get-Open, Intercept, Evaluate-Pass, Pass...
- DEMO

Direct Evolution



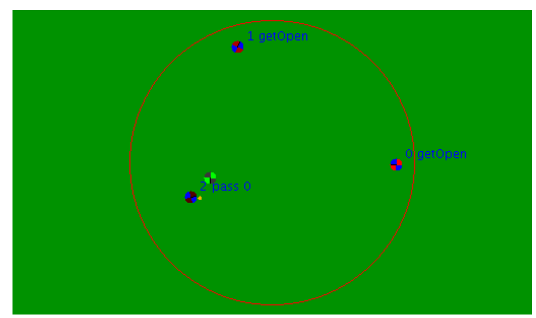
- Mapping sensors directly to actions
  - Difficult to separate behaviors
  - Ineffective combinations evolve
- DEMO

Cooperative Coevolution I



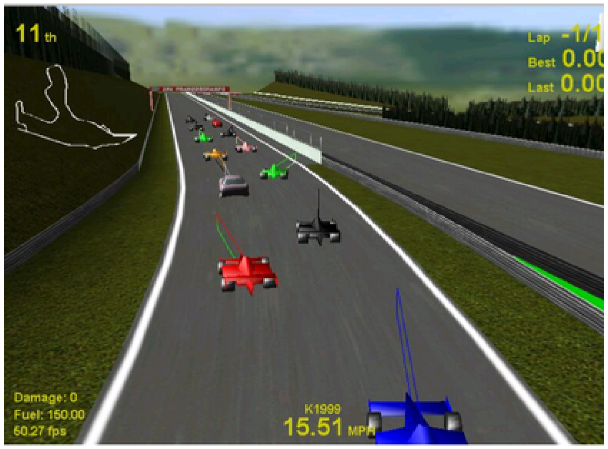
- Evolve multiple actions
  - Each one in a separate network
  - Decision tree to decide on actions

Cooperative Coevolution II



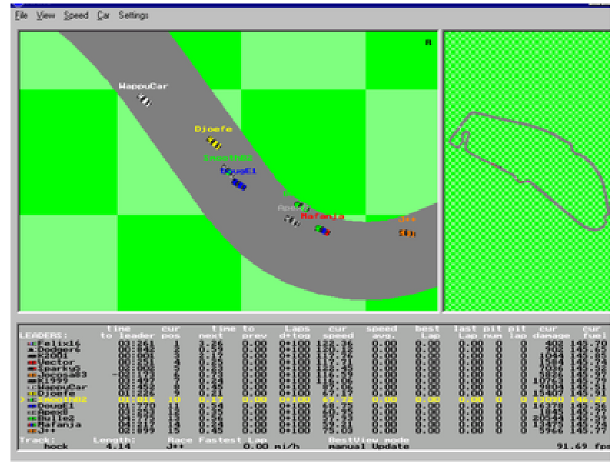
- Networks learn individual tasks
- Learn to anticipate other tasks
  - Lining up for a pass
- Cooperative coevolution of composite behavior
- DEMO

Driving and Collision Warning



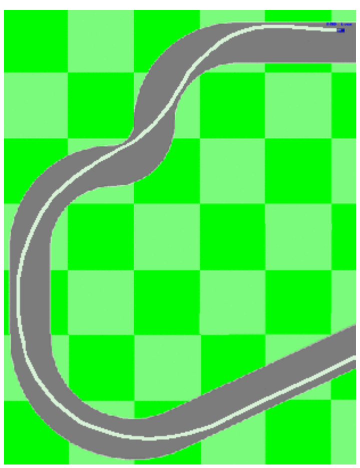
- Goal: evolve a collision warning system
  - Looking over the driver's shoulder
  - Adapting to drivers and conditions
  - Collaboration with Toyota[42]

The RARS Domain



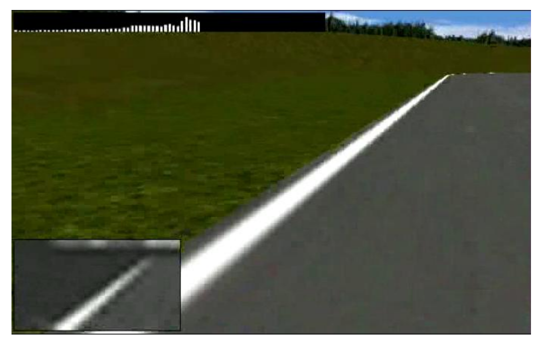
- RARS: Robot Auto Racing Simulator
  - Internet racing community
  - Hand-designed cars and drivers
  - First step towards real traffic

Evolving Good Drivers



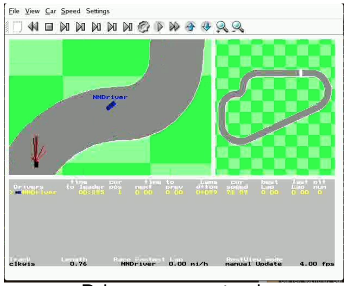
- Evolving to drive fast without crashing
  (off road, obstacles)
- An interesting challenge of its own[104]
- Discovers optimal driving strategies
  (e.g. how to take curves)
- Works from range-finder & radar inputs
- Works from raw visual inputs
  (20×14 grayscale)

Evolving Warnings

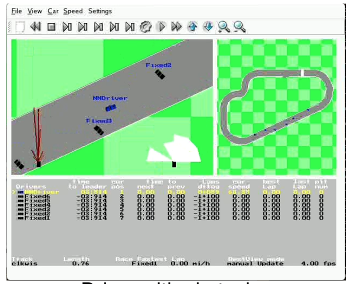


- Evolving to estimate probability of crash
- Predicts based on subtle cues (e.g. skidding off the road)
- Compensates for disabled drivers
- Human drivers learn to drive with it!

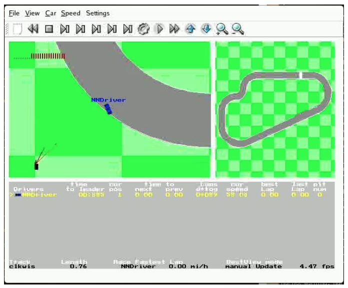
RARS Driving and Collision Warning Demos



Driver on open track



Driver with obstacles



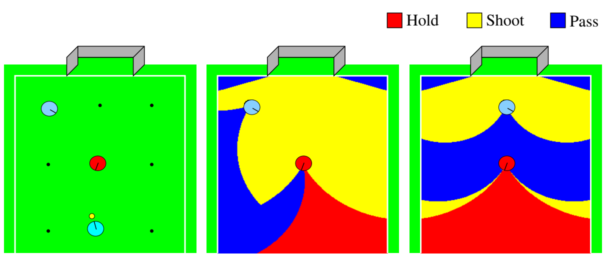
Warning of skidding

Warning of other cars

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

Challenge: Evolving A High-Level Strategy

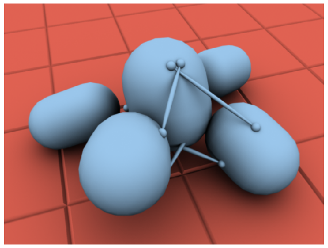


- Instead of continuous control, discrete decision making
  - Choose different behaviors at different times
- Difficult because the decisions are *fractured*[40,41]
  - Optimal behavior changes frequently and discontinuosly
- Need to evolve local decisions
  - E.g. RBF-NEAT, Cascade-NEAT...

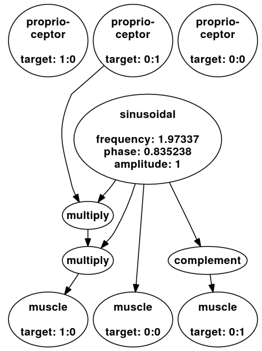
Applications to Artificial Life

- Gaining insight into neural structure
  - E.g. evolving a command neuron[2,39,80]
- Understanding animal behaviors
  - Signaling, herding, hunting...[69,73,74,75,106,115,116,125]

Body-Brain Coevolution



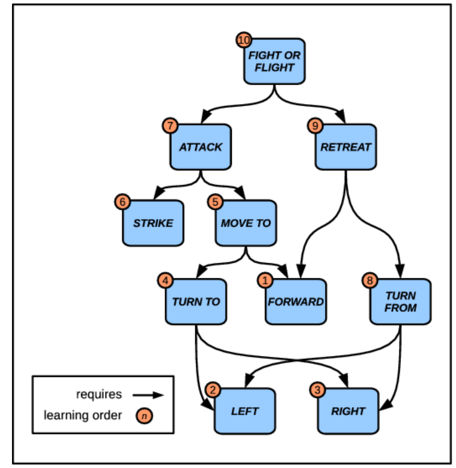
Body



Brain

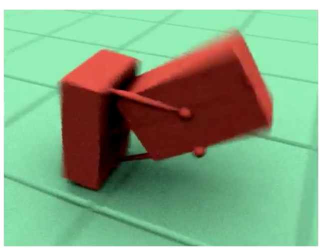
- Evolved Virtual Creatures[48,49,91]
  - Body: Blocks, muscles, joints, sensors
  - Brain: A neural network (with general nodes)
  - Evolved together in a physical simulation
- Syllabus, Encapsulation, Pandemodium

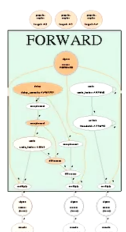
Syllabus



- Constructed by hand; body and brain evolved together

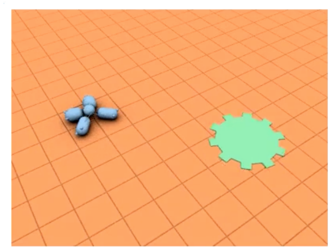
Encapsulation

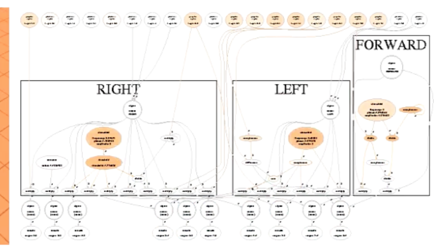




- Once evolved, a trigger node is added
- DEMO

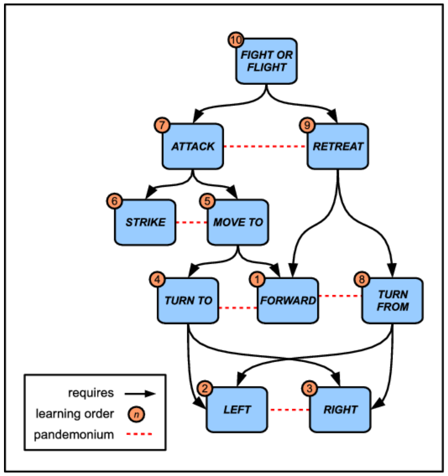
Pandemonium





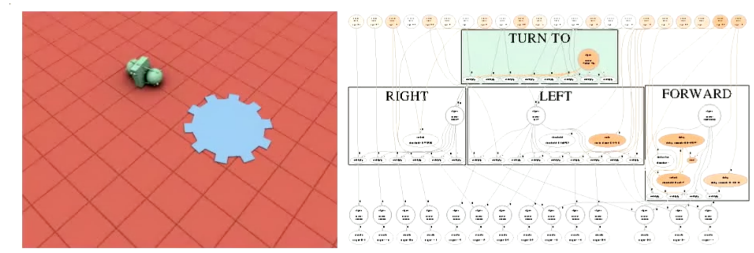
- Conflicting behaviors: Highest trigger wins
- DEMO

Evolving Fight-or-Flight Behavior



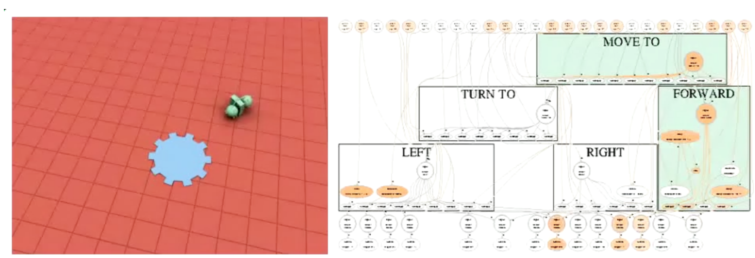
- Step-by-step construction of complex behavior
- Primitives and three levels of complexity
- DEMOS

Turn to Light



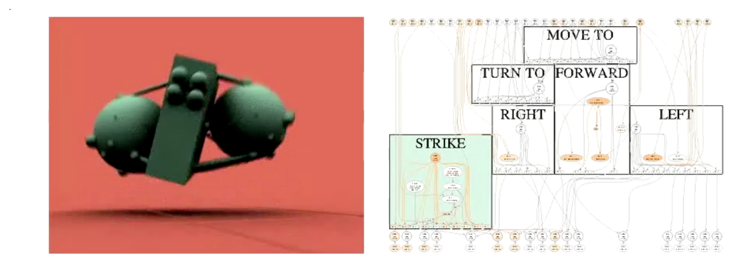
- First level of complexity
- Selecting between alternative primitives

Move to light



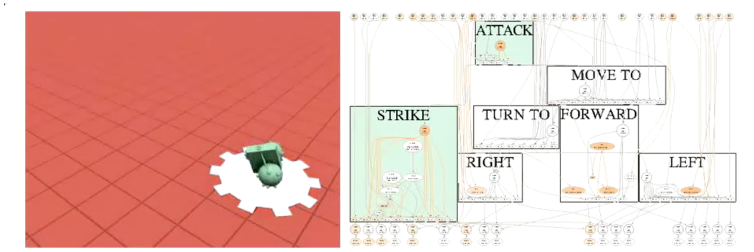
- First level of complexity (Sims 1994)
- Selecting between alternative primitives

Strike



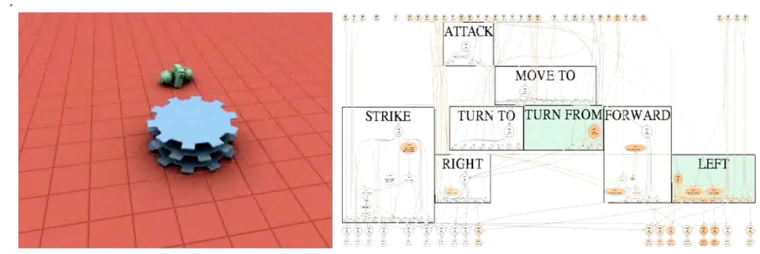
- Alternative behavior primitive

Attack



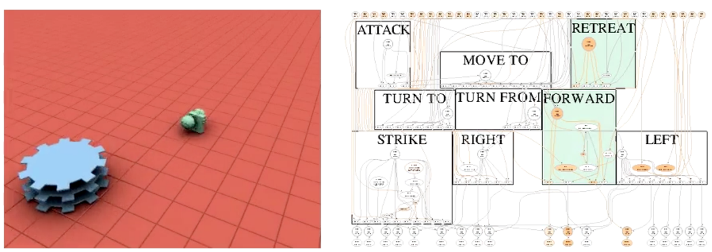
- Second level of complexity (beyond Sims and others)

Turn from Light



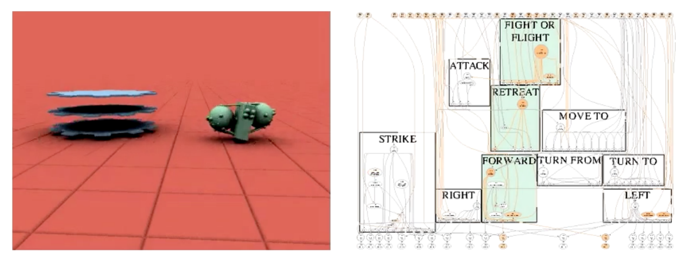
- Alternative first-level behavior

Retreat



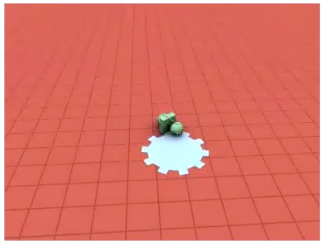
- Alternative second-level behavior

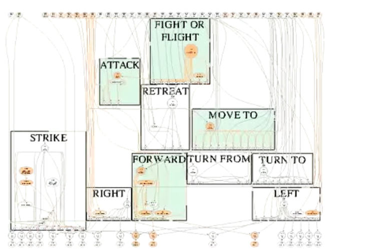
Fight or Flight



- Third level of complexity

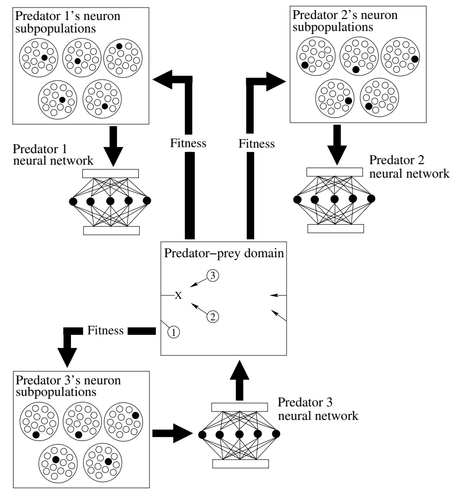
Insight: Body/Brain Coevolution





- Evolving body and brain together poses strong constraints
  - Behavior appears believable
  - Worked well also in BotPrize (Turing test for game bots)
- What about constraints from the environment?

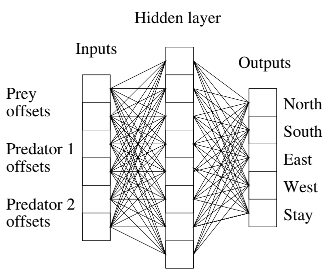
Emergence of Cooperation

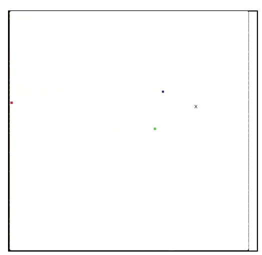


Multi-Agent ESP[124]

- Natural extension of ESP to multiple networks
- Each network constructed from its own subpopulations
- Example: A team catching a fast prey
  - 3 predators, toroidal world
  - Prey as fast, runs away from nearest agent
  - Need to coordinate an approach

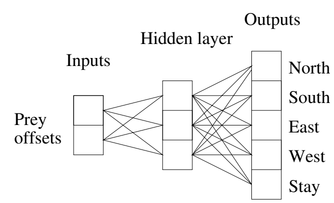
Communication-based Cooperation

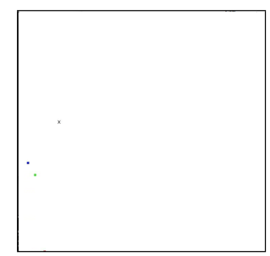




- Individual controllers for each agent
  - Observe the prey and the other predators
  - Develop flexible roles
- Distributed control works better than central control
  - Subtasking through global fitness
- DEMO

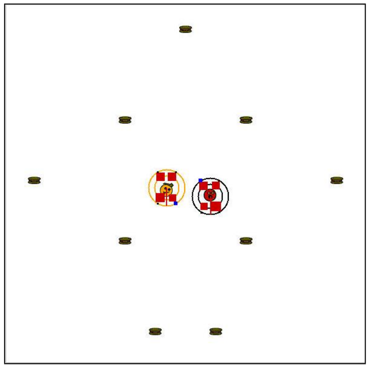
Role-Based Cooperation





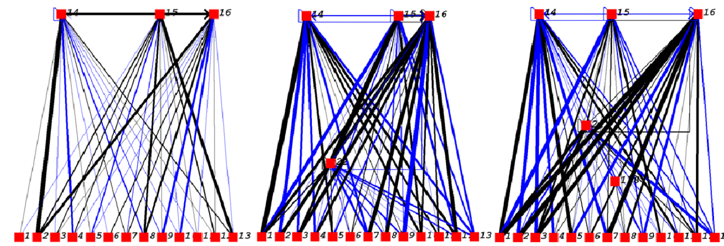
- Each controller only sees the prey
  - Coordination through stigmergy
  - Develop efficient roles
- DEMO
- More effective than communication-based
  - Works like a well-practiced soccer team!
- Multiagent NE powerful in discovering team behaviors

Competitive Coevolution



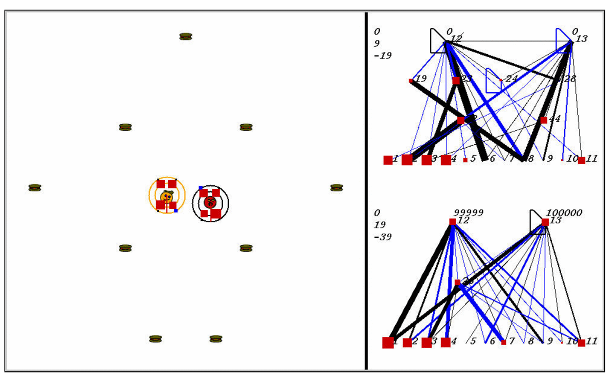
- Evolution requires an opponent to beat
- Such opponents are not always available
- Co-evolve two populations to outdo each other
- How to maintain an arms race?[59]

Competitive Coevolution with NEAT



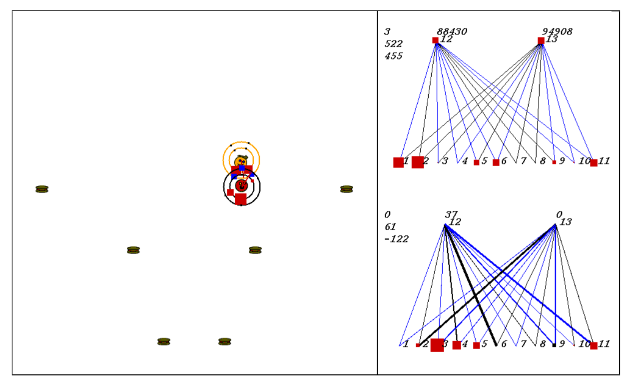
- Complexification elaborates instead of alters
  - Adding more complexity to existing behaviors
- Can establish a coevolutionary arms race
  - Two populations continually outdo each other
  - Absolute progress, not just tricks

Robot Duel Domain



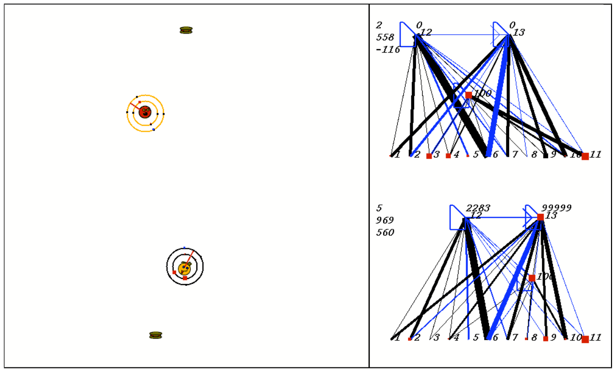
- Two Khepera-like robots forage, pursue, evade[99]
  - Collect food to gain energy
  - Win by crashing to a weaker robot

Early Strategies



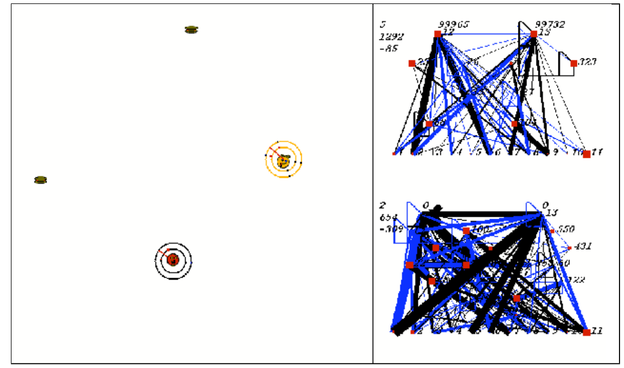
- 1. Rest and let opponent waste energy
- 2. Mainly forage, occasionally crash by accident
- Difficult to switch between tasks
- DEMO

Mature Strategies



- Recurrent hidden node allows switching between tasks
- Collect food to gain energy; rest to save energy
- Difficult to predict energy at contact
- DEMO

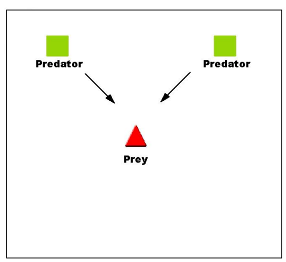
A Sophisticated Strategy



- 1. Split & recurrent connections predict crash outcome
- 2. Complex structure to anticipate opponent behavior
  - “Fake” a rest; entice opponent to forage far away
  - Win by making a dash to last piece
- Complexification of strategy
- DEMO

Coevolution of Behavior

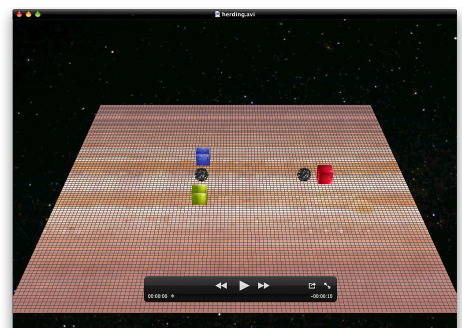
Natural predators and prey



Formalization of behavior

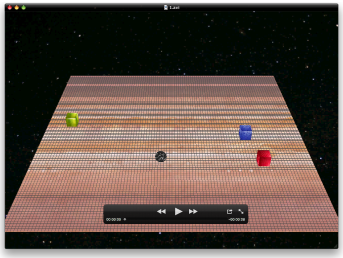
- Complex cooperation observed in pursuit and evasion
  - Motivated by biology, esp. hyenas vs. zebras (Kay Holekamp, MSU)
  - Largely innate, possible to see behaviors and their evolution
- Such behaviors evolve together, in coevolutionary environment
  - Simultaneous competitive and cooperative coevolution[73,75]

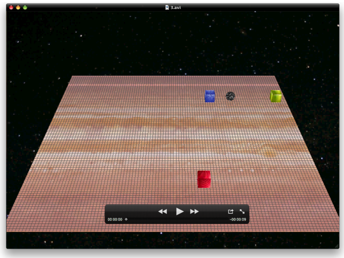
Experimental Setup

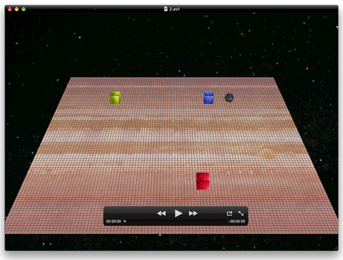


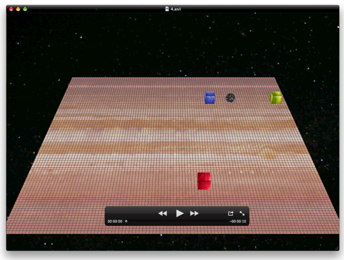
- Toroidal grid world
- Predators, prey move with same speed in 4 directions
- No direct communication between team members
  - Communication still possible through stigmergy
- Does a coevolutionary arms race result?

Predator-Prey Arms Race Demo I

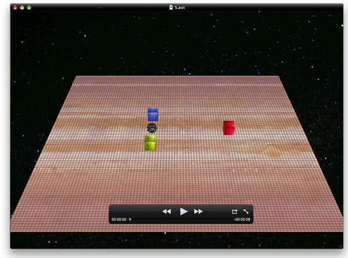

50-75: Single predator catches prey

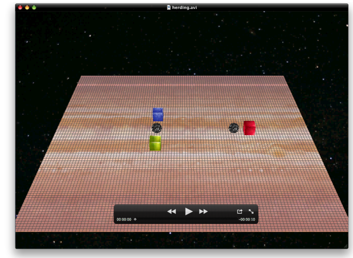

75-100: Prey evades by circling

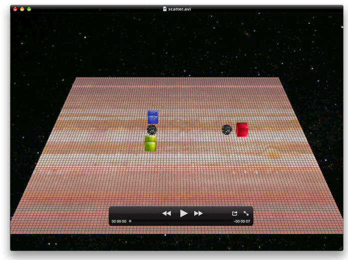

100-150: Two predators cooperate


150-180: Prey baits and escapes

Predator-Prey Arms Race Demo II


180-200: All predators cooperate


200-250: Predators herd two prey


250-300: Prey evade by scattering

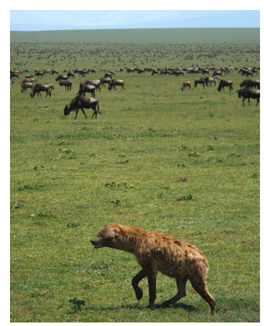
Complex behaviors don't evolve in a vacuum

- Result from coevolutionary arms race
- Embedded in a changing environment

Open Questions

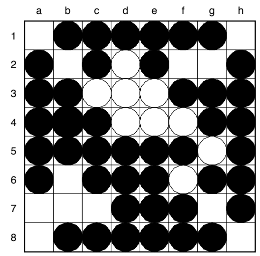
- Role of communication
  - Stigmergy vs. direct communication in hunting[125]
  - Quorum sensing in e.g. confronting lions
- Role of rankings
  - Efficient selection when evaluation is costly?
- Role of individual vs. team rewards
- Can lead to general computational insights

Bigger Questions



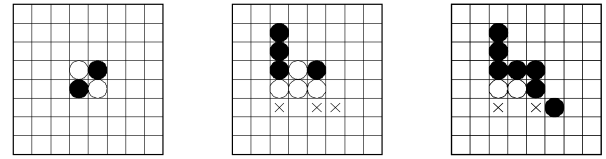
- Gaining insight into cognitive architectures
  - Executive, perception, emotion, memory
- Emergence of language, learning, social structures
- May require overcoming deception
  - Through speciation, niching in nature[46]
  - Through novelty search in computation?[45]

Applications to Games



- Good research platform[57]
  - Controlled domains, clear performance, safe
  - Economically important; training games possible
- Board games: beyond limits of search
  - Evaluation functions in checkers, chess[10,21,22]
  - Filtering information in go, othello[61,100]
  - Opponent modeling in poker[53]

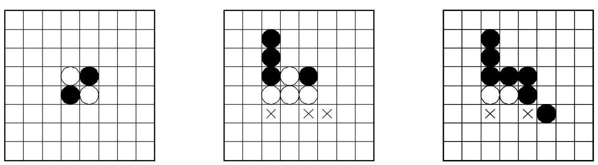
Discovering Novel Strategies in Othello



(a) (b) (c)

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

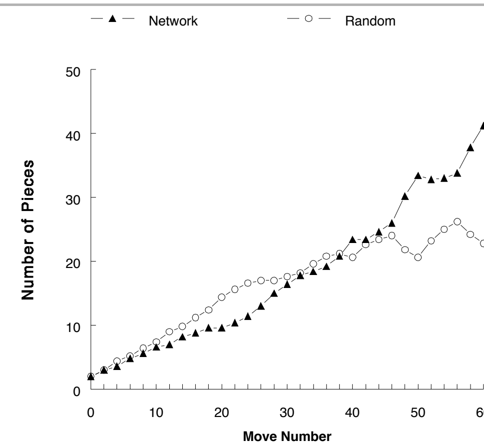
Strategies in Othello



(a) (b) (c)

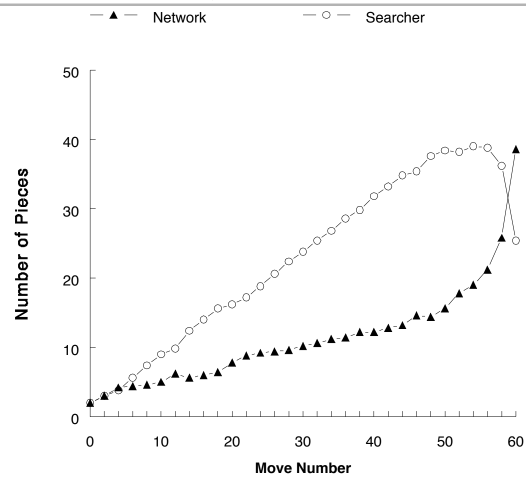
- Positional
  - Number of pieces and their positions
  - Typical novice strategy
- Mobility
  - Number of available moves: force a bad move
  - Much more powerful, but counterintuitive
  - Discovered in 1970's in Japan

Evolving Against a Random Player



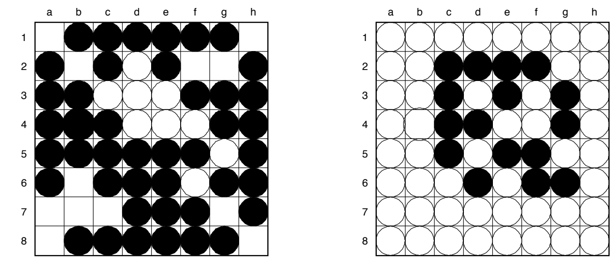
- Network sees the board, suggests moves by ranking[62]
- Networks maximize piece counts throughout the game
- A positional strategy emerges
- Achieved 97% winning percentage

Evolving Against an α-β Program



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

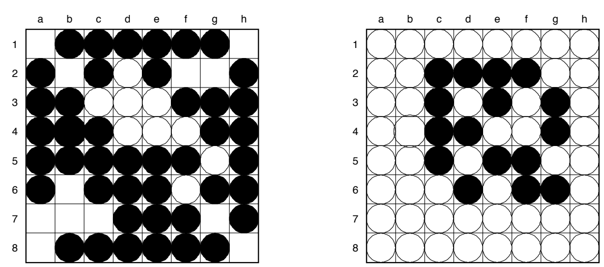
Example game



(a) (b)

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

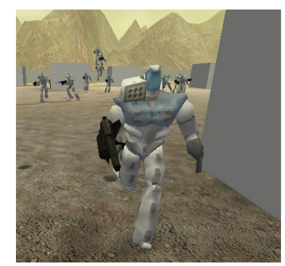
Discovering Novel Strategies



(a) (b)

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

Video Games

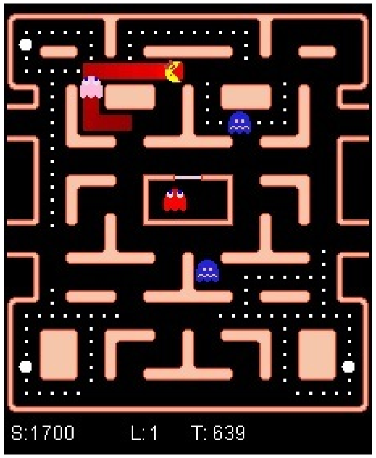


- Economically and socially important
- GOFAI does not work well
  - Embedded, real-time, noisy, multiagent, changing
  - Adaptation a major component
- Possibly research catalyst for CI
  - Like board games were for GOFAI in the 1980s

Video Games II

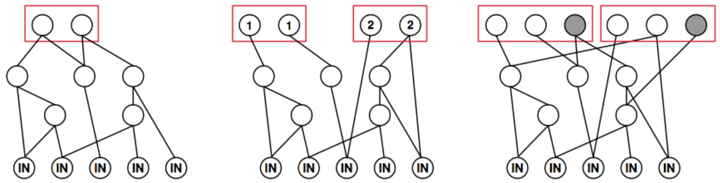
- Can be used to build “mods” to existing games
  - Adapting characters, assistants, tools
- Can also be used to build new games
  - New genre: Machine Learning game

Challenge 1: Evolving Multimodal Behavior



- Agents perform many different tasks
  - E.g. eat pills, avoid ghosts, eat powerpills, eat ghosts
  - Sometimes clearly separate in time
  - Sometimes multiple tasks at once
- How can we evolve them into a single network?

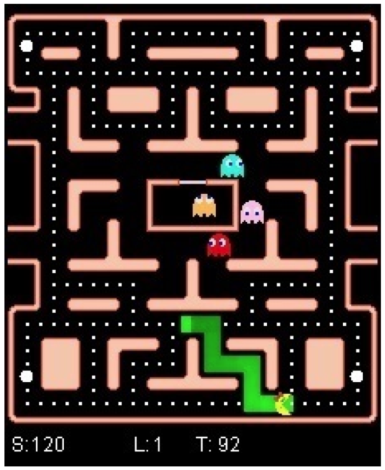
MM-NEAT: Modular Multiobjective Approach



(a) Single-module Network (b) Multitask Network (c) Preference Neuron Network

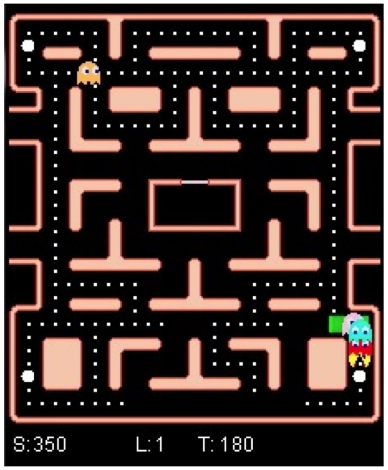
- Evolution discovers modules and when to use them
  - Vs. human-designed division with multitasking
- Multiple modules with preference neurons[85,86]
  - Modules implement different behaviors
  - Preference neurons used to choose among them
  - Module-mutation adds new modules
- Evolved towards multiple objectives
  - Correspond to dimensions of game play
  - E.g. pills and ghosts in Ms. Pac-Man

Human-Designed Task Division



- Multitask approach
  - One module for threat ghosts
  - Another module for edible ghosts
  - Works ok, but...
  - DEMO

Evolution-Discovered Task Division



- One module used 95% of the time
  - Eat pills, avoid ghosts, chase ghosts
  - Different behaviors with a common base
- A second module 5% of the time
  - Luring ghosts near a power pill
  - Escaping from tight spaces
- A different multimodal perspective
- Not as obvious, but more powerful
- DEMO

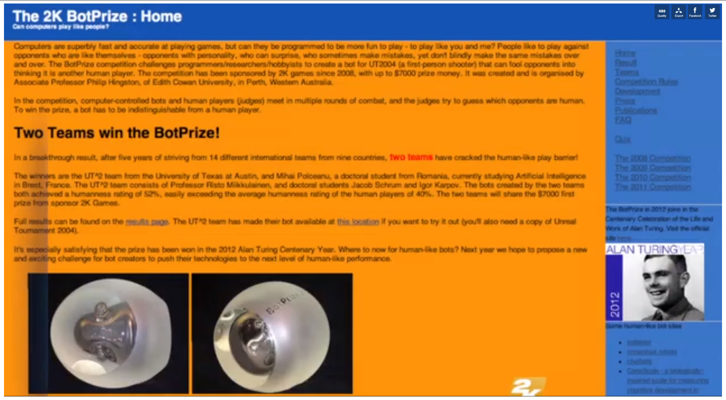
Challenge 2: Evolving Humanlike Behavior

- Botprize competition, 2007-2012
  - Turing Test for game bots ($10,000 prize)
- Three players in Unreal Tournament 2004:
  - Human confederate: tries to win
  - Software bot: pretends to be human
  - Human judge: tries to tell them apart!

Evolving an Unreal Bot

- Evolve effective fighting behavior
  - Human-like with resource limitations (speed, accuracy...)
- Also scripts & learning from humans (unstuck, wandering...)
- 2007-2011: bots 25-30% vs. humans 35-80% human
- 6/2012 best bot better than 50% of the humans
- 9/2012...?

Success!!!

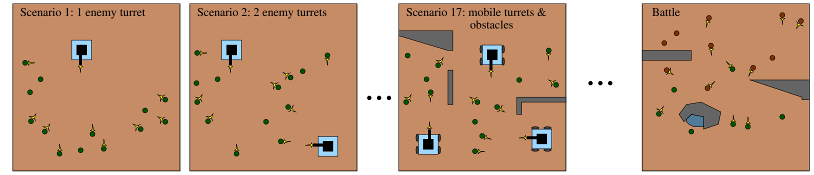


- In 2012, two teams reach the 50% mark!
- Fascinating challenges remain:
  - Judges can still differentiate in seconds
  - Judges lay cognitive, high-level traps
  - Team competition: collaboration as well
- DEMO

A New Genre: Machine Learning Games

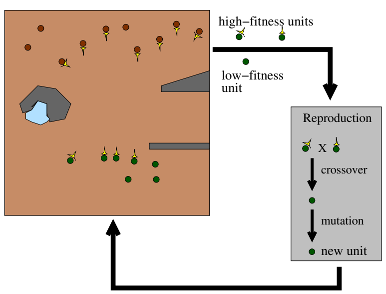
- E.g. NERO
  - Goal: to show that machine learning games are viable
  - Professionally produced by *Digital Media Collaboratory*, UTAustin
  - Developed mostly by volunteer undergraduates

NERO Gameplay



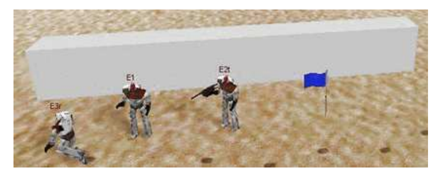
- Teams of agents trained to battle each other
  - Player trains agents through excercises
  - Agents evolve in real time
  - Agents and player collaborate in battle
- New genre: Learning *is* the game[33,96]
  - Challenging platform for reinforcement learning
  - Real time, open ended, requires discovery
- Try it out:
  - Available for download at `http://nerogame.org`
  - Open source research platform version at `github.com/nnrg/opennero/wiki`

Real-time NEAT



- A parallel, continuous version of NEAT[96]
- 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

NERO Training Demos


Approach Enemy


Switch to Avoid


Avoid, first-person


Maze Running

NERO Battle Demo

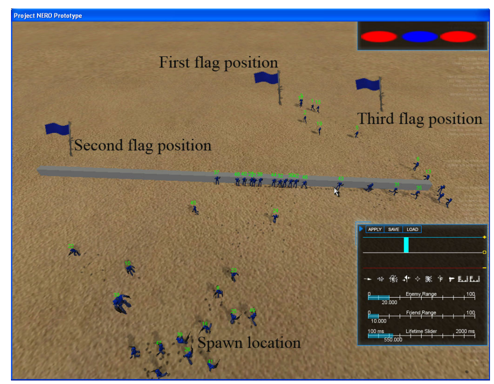

Aggressive vs. Avoidant


Teams of three

Combining Evolution With Human Knowledge

- Given a problem, NE discovers a solution by exploring
  - Sometimes you already know (roughly) what works
  - Sometimes random initial behavior is not acceptable
- How can game developer's knowledge be utilized?
  - By learning from examples (as in BotPrize)[8]
  - By incorporating rules[13,126]

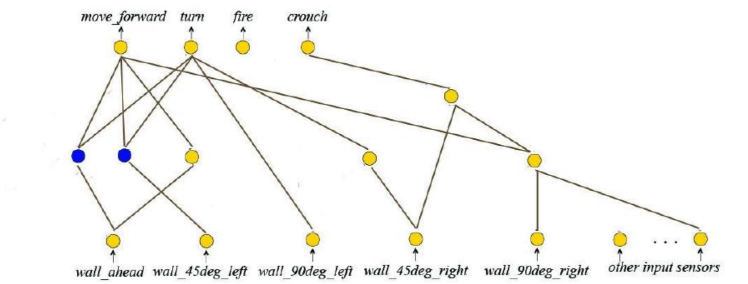
Incorporating Rules into NE I



E.g. how to go around a wall in NERO

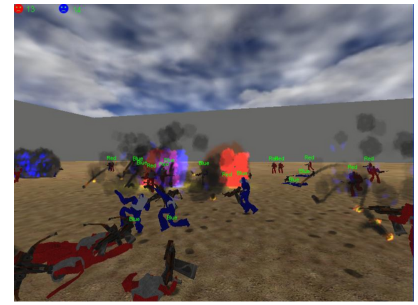
- Specify as a rule:
  - *wall_ahead: move_forward, turn_right*
  - *wall_45deg_left, move_forward, turn_right_slightly*
- Convert into a network with KBANN[54]

Incorporating Rules into NE II



- KBANN network added to NEAT networks
  - Treated as complexification
  - Continues to evolve
  - If advice is useful, it stays
- Initial behaviors, on-line advice
- Injecting human knowledge as rules
- DEMO

Lessons from NERO



- NEAT is a strong method for real-time adaptation
  - Complex team behaviors can be constructed
  - Novel strategies can be discovered
- Problem solving with human guidance
- NE makes a new genre of games possible!

Numerous Other Applications

- Creating art, music, dance...[11,17,35,88]
- Theorem proving[16]
- Time-series prediction[55]
- Computer system optimization[26]
- Manufacturing optimization[31]
- Process control optimization[113,114]
- Game strategy optimization[4]
- Measuring top quark mass[120]
- 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 needs to be developed
  - Indirect encodings
  - Learning and evolution
  - Knowledge, interaction, novelty

Further Material

- Slides (including the bibliography) available at www.cs.utexas.edu/users/risto/talks/ne-tutorial
- Demos in this talk are at www.cs.utexas.edu/users/risto/talks/ne-tutorial and many more at nn.cs.utexas.edu
- A Scholarpedia article on Neuroevolution is at www.scholarpedia.org/article/Neuroevolution
- A step-by-step neuroevolution exercise (evolving behavior in the NERO game) is at www.cs.utexas.edu/users/risto/talks/ne-tutorial

Bibliography I

[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] E. Bahceci and R. Miikkulainen, Evolving strategies for social innovation games, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2015)*, Madrid, Spain (July 2015).

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

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

[7] 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).

[8] 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*, AAAI Press, Menlo Park, CA (2007).

[9] D. J. Chalmers, The evolution of learning: An experiment in genetic connectionism, in: Touretzky et al.[105], 81–90.

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

[11] 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).

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

Bibliography II

[13] R. Cornelius, K. O. Stanley, and R. Miikkulainen, Constructing adaptive AI using knowledge-based neuroevolution, in: *AI Game Programming Wisdom 3*, S. Rabin, ed., 693–708, Charles River Media, Revere, MA (2006).

[14] 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).

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

[16] 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).

[17] 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).

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

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

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

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

[22] 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).

[23] 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).

Bibliography III

[24] 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).

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

[26] 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).

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

[28] 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).

[29] F. Gomez and R. Miikkulainen, Transfer of neuroevolved controllers in unstable domains, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, Springer, Berlin (2004).

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

[31] 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).

[32] 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).

[33] 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).

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

[35] 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).

Bibliography IV

[36] 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).

[37] 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).

[38] A. Jain, A. Subramoney, and R. Miikkulainen, Task decomposition with neuroevolution in extended predator-prey domain, in: *Proceedings of Thirteenth International Conference on the Synthesis and Simulation of Living Systems*, East Lansing, MI, USA (2012).

[39] 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).

[40] N. Kohl and R. Miikkulainen, Evolving neural networks for fractured domains, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, 1405–1412 (July 2008).

[41] N. Kohl and R. Miikkulainen, Evolving neural networks for strategic decision-making problems, *Neural Networks*, 22:326–337 (2009).

[42] 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).

[43] R. Koppejan and S. Whiteson, Neuroevolutionary reinforcement learning for generalized control of simulated helicopters, *Evolutionary Intelligence*, 4:219–241 (2011).

[44] J. Lehman and R. Miikkulainen, Effective diversity maintenance in deceptive domains, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2013).

[45] J. Lehman and R. Miikkulainen, Overcoming deception in evolution of cognitive behaviors, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014)*, Vancouver, BC, Canada (July 2014).

[46] J. Lehman and R. Miikkulainen, Enhancing divergent search through extinction events, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2015)*, Madrid, Spain (July 2015).

[47] J. Lehman and K. O. Stanley, Abandoning objectives: Evolution through the search for novelty alone, *Evolutionary Computation*, 2011:189–223 (2010).

Bibliography V

[48] D. Lessin, D. Fussell, and R. Miikkulainen, Open-ended behavioral complexity for evolved virtual creatures, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2013).

[49] D. Lessin, D. Fussell, and R. Miikkulainen, Trading control intelligence for physical intelligence: Muscle drives in evolved virtual creatures, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014)*, Vancouver, BC, Canada (July 2014).

[50] J. Z. Liang and R. Miikkulainen, Evolutionary bilevel optimization for complex control tasks, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2015)*, Madrid, Spain (July 2015).

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

[52] A. Lockett and R. Miikkulainen, Neuroannealing: Martingale-driven learning for neural network, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2013).

[53] 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).

[54] R. Maclin and J. W. Shavlik, Creating advice-taking reinforcement learners, *Machine Learning*, 22(1-3):251–281 (1996).

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

[56] 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.

[57] 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).

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

[59] G. Monroy, K. O. Stanley, and R. Miikkulainen, Coevolution of neural networks using a layered pareto archive, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2006).

Bibliography VI

[60] 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).

[61] 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.

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

[63] 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).

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

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

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

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

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

[69] S. Nolfi and M. Mirolli, eds., *Evolution of Communication and Language in Embodied Agents*, Springer, Berlin (2010).

[70] 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).

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

Bibliography VII

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

[73] P. Rajagopalan, A. Rawal, R. Miikkulainen, M. A. Wiseman, and K. E. Holekamp, The role of reward structure, coordination mechanism and net return in the evolution of cooperation, in: *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG 2011)*, Seoul, South Korea (2011).

[74] A. Rawal, P. Rajagopalan, K. E. Holekamp, and R. Miikkulainen, Evolution of a communication code in cooperative tasks, in: *Proceedings of Thirteenth International Conference on the Synthesis and Simulation of Living Systems (ALife'2013)*, East Lansing, MI, USA (2012).

[75] A. Rawal, P. Rajagopalan, and R. Miikkulainen, Constructing competitive and cooperative agent behavior using coevolution, in: *IEEE Conference on Computational Intelligence and Games (CIG 2010)*, Copenhagen, Denmark (2010).

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

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

[78] C. D. Rosin and R. K. Belew, New methods for competitive evolution, *Evolutionary Computation*, 5 (1997).

[79] 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).

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

[81] 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).

[82] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks*, 61:85–117 (2015).

Bibliography VIII

[83] 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).

[84] 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).

[85] J. Schrum and R. Miikkulainen, Evolving multimodal behavior with modular neural networks in ms. pac-man, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014)*, Vancouver, BC, Canada (July 2014).

[86] J. Schrum and R. Miikkulainen, Solving interleaved and blended sequential decision-making problems through modular neuroevolution, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2015)*, Madrid, Spain (July 2015).

[87] J. Secretan, N. Beato, D. B. D'Ambrosio, A. Rodriguez, A. Campbell, J. T. Folsom-Kovarik, and K. O. Stanley, Picbreeder: A case study in collaborative evolutionary exploration of design space, *Evolutionary Computation*, 19:345–371 (2011).

[88] 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).

[89] 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).

[90] 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).

[91] K. Sims, Evolving 3D morphology and behavior by competition, in: *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems (Artificial Life IV)*, R. A. Brooks and P. Maes, eds., 28–39, MIT Press, Cambridge, MA (1994).

Bibliography IX

[92] A. Sinha, P. Malo, P. Xu, and K. Deb, A bilevel optimization approach to automated parameter tuning, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014)*, Vancouver, BC, Canada (July 2014).

[93] 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).

[94] 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).

[95] K. O. Stanley, B. D. Bryant, and R. Miikkulainen, Evolving adaptive neural networks with and without adaptive synapses, in: *Proceedings of the 2003 Congress on Evolutionary Computation*, IEEE, Piscataway, NJ (2003).

[96] 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).

[97] K. O. Stanley and R. Miikkulainen, Evolving Neural Networks Through Augmenting Topologies, *Evolutionary Computation*, 10:99–127 (2002).

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

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

[100] 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).

[101] 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).

[102] W. Tansey, E. Feasley, and R. Miikkulainen, Accelerating evolution via egalitarian social learning, in: *Proceedings of the 14th Annual Genetic and Evolutionary Computation Conference (GECCO 2012)*, Philadelphia, Pennsylvania, USA (July 2012).

[103] 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).

Bibliography X

[104] 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).

[105] D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, eds., *Proceedings of the 1990 Connectionist Models Summer School*, San Francisco: Morgan Kaufmann (1990).

[106] E. Tuci, An investigation of the evolutionary origin of reciprocal communication using simulated autonomous agents, *Biological Cybernetics*, 101:183–199 (2009).

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

[108] V. Valsalam, J. Hiller, R. MacCurdy, H. Lipson, and R. Miikkulainen, Constructing controllers for physical multilegged robots using the enso neuroevolution approach, *Evolutionary Intelligence*, 14:303–331 (2013).

[109] V. Valsalam and R. Miikkulainen, Evolving symmetry for modular system design, *IEEE Transactions on Evolutionary Computation*, 15:368–386 (2011).

[110] V. K. Valsalam, J. A. Bednar, and R. Miikkulainen, Constructing good learners using evolved pattern generators, in: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2005*, H.-G. Beyer et al., eds., 11–18, New York: ACM (2005).

[111] 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).

[112] V. K. Valsalam and R. Miikkulainen, Evolving symmetric and modular neural networks for distributed control, in: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2009*, 731–738, ACM, New York, NY, USA (2009).

[113] 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., San Francisco: Morgan Kaufmann (2002).

[114] A. van Eck Conradie, R. Miikkulainen, and C. Aldrich, Intelligent process control utilizing symbiotic memetic neuro-evolution, in: *Proceedings of the 2002 Congress on Evolutionary Computation* (2002).

[115] 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).

[116] 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).

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

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

[119] S. Whiteson, P. Stone, K. O. Stanley, R. Miikkulainen, and N. Kohl, Automatic Feature Selection in Neuroevolution, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (2005).

[120] 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).

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

[122] A. P. Wieland, Evolving controls for unstable systems, in: Touretzky et al. [105], 91–102.

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

[124] 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).

[125] C. H. Yong and R. Miikkulainen, Coevolution of role-based cooperation in multi-agent systems, *IEEE Transactions on Autonomous Mental Development*, 1:170–186 (2010).

[126] 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).