Introductory Tutorial on Coevolution

Edwin de Jong
Utrecht University
Utrecht, The Netherlands
dejong@cs.uu.nl

Kenneth Stanley
University of Central Florida
Orlando FL USA
kstanley@cs.ucf.edu

R. Paul Wiegand
University of Central Florida
Orlando FL USA
wiegand@ist.ucf.edu

GECCO 2007

Outline

Background

What is Coevolution?

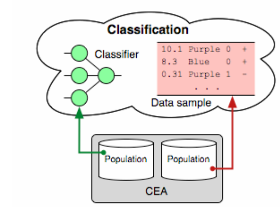
Form of evolutionary computation in which the fitness evaluation is based on interactions between multiple individuals

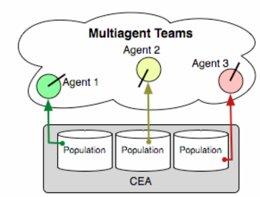
- An individual's ranking in a population can change depending on other individuals
- Coevolutionary fitness is *absolute* or *subjective*
- Traditional EC fitness is *relative* or *objective*

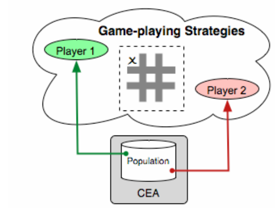
Background

Why use Coevolution?

- Large (infinite) search spaces
- No objective measures exist
- Objective measure difficult to formalize or unknown
- Certain types of structure in search space







GECCO'07, July 7–11, 2007, London, England, United Kingdom.
ACM 978-1-59593-698-1/07/0007.

Background

Historical Examples: Early Work

- Samuel 1959, 1967
  - learning checkers through self-play
- Barricelli 1963; Reed, Toombs, & Barricelli 1967
  - TacTix (game similar to Nim)
- Axelrod 1987
  - iterated prisoner's dilemma

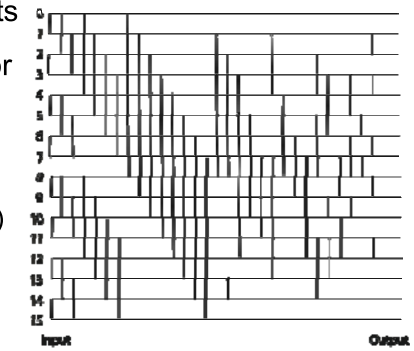
Background

Historical Examples: Sorting Networks

Hillis 1990

- Learner-teacher paradigm
- Coevolves sorting networks against inputs
- Obtains 61-comparator network

(just one more than best known for 16-input problem)



Background

Historical Examples: Virtual Creatures

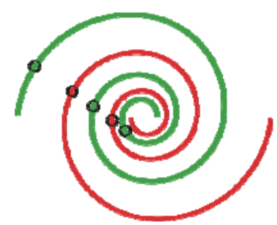
Sims 1994

- Virtual creatures in simulated physics environment
- Pair-wise competitions to gain control over a cube in the middle of the arena
- Coevolution of agent morphology and control
- Variety of interesting body plans and behaviors obtained

Background

Historical Examples: Intertwined Spirals

Juillé & Pollack 1996



- Difficult classification problem
- Motivated by study of neural networks
- 194 data points to classify

- Coevolves genetic-program classifiers, where payoff to Player *i* is:
  - G(i, j) = #points "covered" by Player *i* that are not covered by Player *j*
- Finds modular solutions to problem:
  - Divides space, solves each region independently

Background

Historical Examples: Checkers

Chellapilla & Fogel 1999 & 2000

- Coevolves weights of neural network used to evaluate game boards
- Combined with four-ply lookahead
- Initial work achieved “Class-A” designation
- Subsequent work produced “Expert” level
- (Just below Master and Grand Master)

Coevolutionary Algorithms

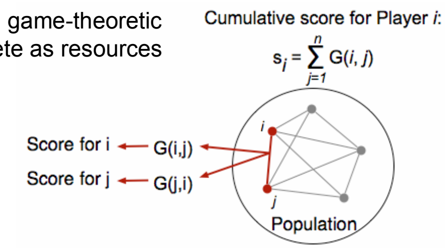
Coevolutionary Algorithms (CEAs)

- Very similar to traditional EA methods
  - Individuals encode aspect of potential solutions
  - They are altered during search with genetic operators
  - Search directed by selection based on fitness
- But differ in fundamental ways:
  - Evaluation requires interaction between multiple individuals
  - Interacting individuals may reside in same population or in different populations
  - Evokes notions of cooperation and competition in new ways
  - Some representation issues are unique to coevolution
  - Methods of evaluation are particularly important

Coevolutionary Algorithms

Basic Structure of the CEA

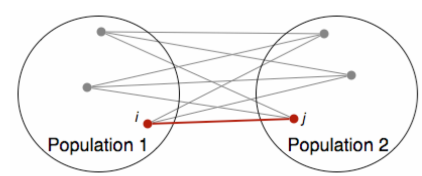
- Single population:
  - Employs a single EA, typically using traditional operators
  - Individuals represent candidate solutions to a problem, as well as “tests” for other individuals
  - Evaluate individual by interacting with other individuals in same population
  - Individuals compete in game-theoretic sense, but also compete as resources for evolution



Coevolutionary Algorithms

Basic Structure of the CEA

- Multiple populations:
  - Multiple EAs, one for each population
  - Individuals may represent a variety of things
  - Evaluate individual by interacting with individuals from other populations
  - Typically individuals in different populations do not compete directly for the EA resources



Coevolutionary Algorithms

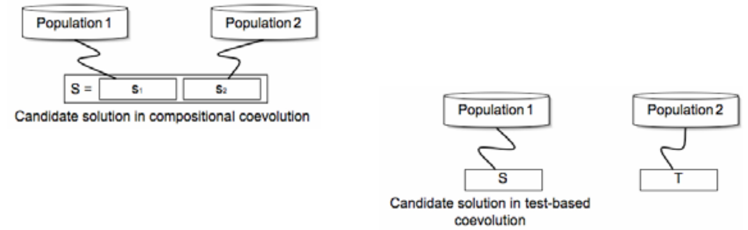
Cooperation & Competition

- Three ways terms used in coevolution
  - Different types of algorithms (e.g., Rosin vs. Potter)
    - Cooperative algorithms are those in which interacting individuals succeed or fail together
    - Competitive algorithms are those in which individuals succeed at the expense of other individuals.
  - Qualitatively observed behaviors of potential solutions (e.g., “tit-for-tat”)
  - Inherent properties of coevolutionary problems (e.g., sorting networks and data sets)

Coevolutionary Algorithms

Compositional & Test-based

- Alternatively, algorithms are distinguished based on how solutions are represented
  - In *compositional coevolution*, solutions are composed of multiple individuals
  - In *test-based coevolution*, individuals represent candidate solutions and/or their tests



Coevolutionary Algorithms

Examples of Algorithm Types

- Compositional/cooperative coevolution:
  - Coevolving a multiagent team responsible for jointly defending a resource (*solution: Team behaviors*)
- Test-based/competitive coevolution:
  - Coevolving a classifier and challenging datasets (*solution: general classifiers*)
- Compositional/competitive coevolution:
  - Coevolving an ecosystem of agents in a market simulation (*solution: ecosystem for analysis*)
- Test-based/cooperative coevolution:
  - Coevolving an autonomous agent for a team with human agents, with other evolving agents simulating human behaviors (*solution: single team member behavior*)

The compositional/test-based distinction is similar to the cooperative/competitive distinction, but not the same

Coevolutionary Algorithms

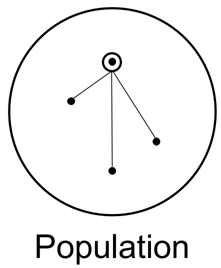
Evaluation: Interaction Patterns

- All vs. all is “canonical” but expensive
- All vs. previous-best
- Tournament
- See Angeline & Pollack 1993, Sims 1994
- Shared sampling [Rosin & Belew 1997]

Coevolutionary Algorithms

Evaluation: All vs. Best

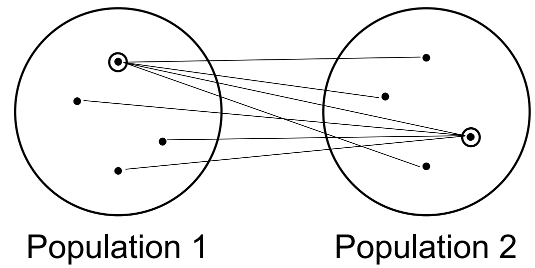
- Individuals interact with "best" individual(s) from previous generation (need elitism)
- Feasible for one or more populations



Coevolutionary Algorithms

Evaluation: All vs. Best

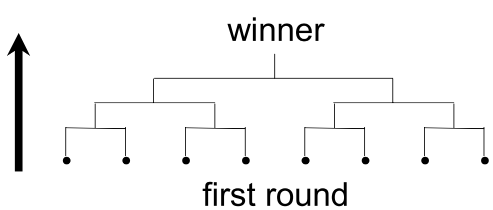
- Individuals interact with "best" individual(s) from previous generation (need elitism)
- Feasible for one or more populations



Coevolutionary Algorithms

Evaluation: Tournament Evaluation

- Pairwise interactions in single-elimination tournament (single-population)
- Each individual's score determined by how far individual progresses in tournament



Coevolutionary Algorithms

Evaluation: Shared Sampling

Rosin 1997

- Purpose to enhance diversity in evaluation
- Based on their Competitive Fitness Sharing method (discussed below)
- Bias sampling of individuals with whom interaction (during evaluation) takes place
- Sample "redundant" individuals less (relative to uniform); "rare" individuals more
- "Redundant" and "rare" determined by similarity in performance

Coevolutionary Algorithms

Representation

- How are solutions encoded?
  - *Test-based coevolution*: individuals represent candidate solutions as well as tests for those solutions
  - *Compositional coevolution*: individuals represent candidate components for a composite or ensemble solution
- What is a population?
  - Group of potential solutions, tests, or components
  - Possibly a collection of *pure strategy* solutions composing a single *mixed strategy* candidate solution

Coevolutionary Algorithms

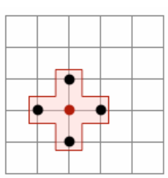
Representation

- How are solutions decomposed?
  - Static, *a priori* decomposition (Potter & De Jong, 1994)
  - Dynamic decomposition (Potter & De Jong, 2001)
  - Decomposition partially determined by CEA (Moriarty & Miikkulainen, 1997)
- How can we take advantage of open-ended evolution?
  - Gradually *complexifying* representational space during search (Stanley & Miikkulainen, 2004)

Coevolutionary Algorithms

Eval. & Rep.: Spatial Coevolution

- Individuals spatially arranged on a lattice
- Individuals interact only with neighbors



Population

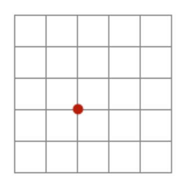
Hillis 1990
Pagie & Hogeweg 1997
Wiegand & Sarma 2004
Mitchell et al. *2006*

Coevolutionary Algorithms

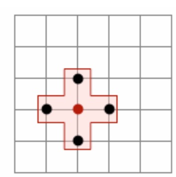
Eval. & Rep.: Spatial Coevolution

- Individuals spatially arranged on a lattice
- Individuals interact only with neighbors
- In two-population system, interact with individuals in corresponding neighborhood of other population

Local neighborhood structure helps maintain population **diversity**, which may help against various pathologies



Population 1



Population 2

Hillis 1990
Pagie & Hogeweg 1997
Wiegand & Sarma 2004
Mitchell et al. *2006*

Pathologies

Pathologies

- Early results sparked interest in coevolution, but various pathologies quickly became evident
- Why coevolution fails to produce desired results is often unclear
- We discuss these pathologies then outline several attempts to remedy them

Pathologies

Two Key Concepts

- Gradient
- Underlying objectives

Concept: Gradient

Pathologies

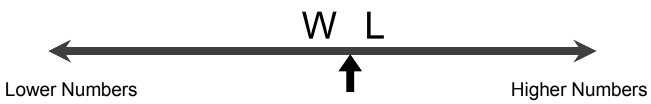
Gradient

- Gradient refers to information provided by evaluation
- The evaluation of individuals depends on other, coevolving individuals
- Ability to distinguish individuals on the basis of their interactions with coevolving individuals
- Roughly, gradient allows an algorithm to tell which individuals appear better

Pathologies

Transitive Numbers Game

- Each player in game is a real number [0, 1]
- Winner is player with higher number



Pathologies

Gradient Leading to “Arms Race”

- Transitive numbers game
- Two populations
- All-vs-all interaction:
  - each in Pop.1 plays all in Pop.2, and *vice versa*
  - players earns one point for each win
- Players reproduce in proportion to total points
- Gaussian mutation: mean = 0, std = 0.01

Pathologies

Disengagement

- The event that gradient is lost—individuals can no longer be distinguished
- Grade inflation: All students receive an ‘A.’
  - Grades no longer useful to distinguish the better students
  - Students and curriculum are “disengaged”

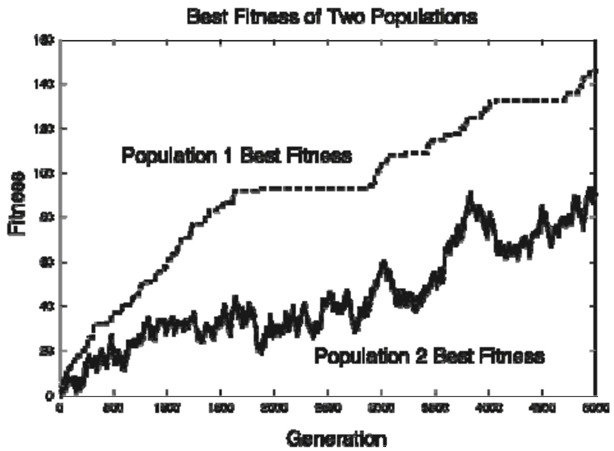
Pathologies

Stalling/Drift

- When disengagement persists over evolutionary time, <u>stalling</u> or <u>drift</u> can occur
- Stalling:
  - If algorithm only replaces individuals with strictly more fit ones (e.g., a hill-climber)...
  - Then, population stops changing
- Drift:
  - If algorithm replaces individuals with others of equal or greater fitness...
  - Then, algorithm will perform a random walk, biased by variation methods

Pathologies

Disengagement, Stalling, and Drift



Pathologies

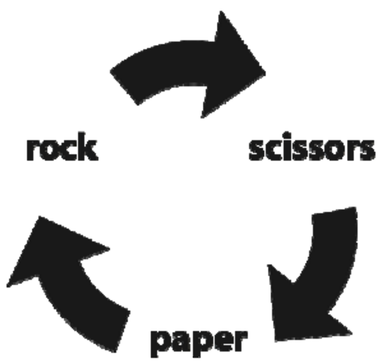
Cycling

- *Cycling* typically refers to an oscillation in some metric of algorithm behavior
- With an offline metric of behavior:
  - May observe performance of coevolved individuals going up and down through evolutionary time
  - May observe that current individuals beat some past individuals but lose to others

Pathologies

Intransitivity

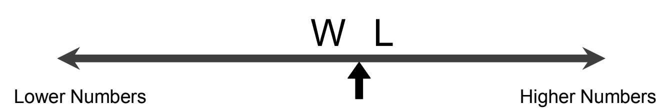
- Intransitivity is a characteristic of a problem domain
- Rock-paper-scissors is a canonical example of an intransitive domain
- Coevolutionary algorithms have been observed to cycle on intransitive domains, but may cycle on any domain



Pathologies

Transitive Numbers Game

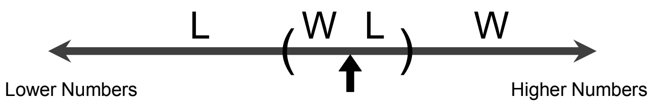
- Each player in game is a real number [0, 1]
- Winner is player with higher number



Pathologies

Cycling/Intransitivity

- Watson's Locally Transitive Intransitive Game



Pathologies

The Red-Queen Effect

van Valen 1973

- In biology:
  - Despite constant genetic change, the extinction probability of a species does not change because of changes in the environment
- In evolutionary computation:
  - Changes that improve the quality of an individual do not increase its selection probability because of changes to other coevolving individuals

Pathologies

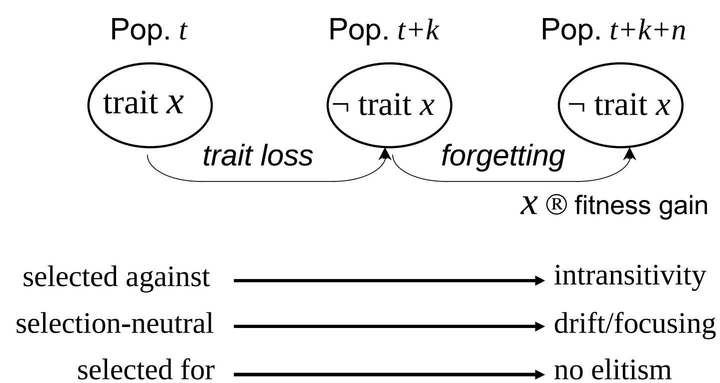
The Red-Queen Effect

van Valen 1973

- Red-Queen Effect prevents us from distinguishing "arms-race" dynamic from:
  - cycling dynamics due to intransitivity
  - algorithm stalling/genetic drift due to disengagement
- New individuals appear as capable as previous ones relative to the present context
- (Unless we have an off-line metric of goodness)

Pathologies

Evolutionary Forgetting

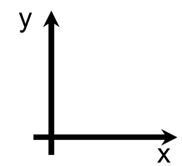


Concept: Underlying Objectives

Pathologies

Underlying Objectives

- Multiobjective algorithms simultaneously optimize several different objective functions
- Consider “capabilities” as objectives
- Similarly, coevolutionary domains might have a <u>set of underlying objectives</u> that must be optimized to produce good individuals



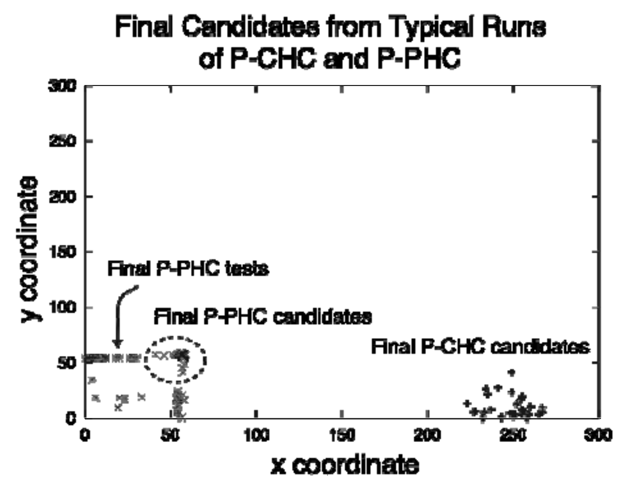
Pathologies

Overspecialization/Focusing

- When individuals improve on some underlying objectives at the expense of others
- For instance, coevolving game players may <u>focus</u> on defeating certain (types of) opponents and not evolve to defeat others

Pathologies

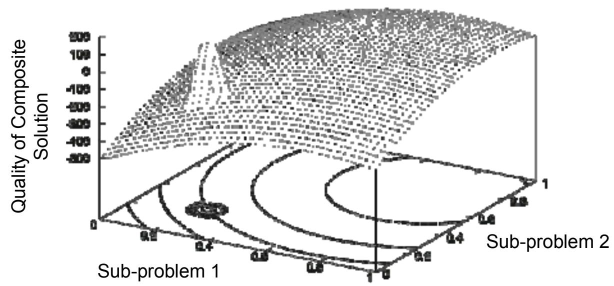
Overspecialization/Focusing



Pathologies

Relative Overgeneralization

- Phenomenon in cooperative coevolution
  - Components/genotypes that perform well with a large number of other individuals...
  - ... are favored over components that are part of an optimal solution



Pathologies

Relationships: Gradient

- Disengagement is a loss of gradient
- Stalling or drifting can result from a lack of gradient which persists through evolutionary time
- Drift may in turn lead to forgetting or overspecialization

Pathologies

Relationships: Underlying Objectives

- Overspecialization is focusing on a subset of the underlying objectives
- Cycling may result from oscillating between two underlying objectives
- Relative overgeneralization can be seen to result from the loss of an underlying objective in Cooperative Coevolution

Remedies

Remedies

- *Forgetting* remedies are typically about distinguishing individuals
  - If individuals cannot be distinguished, some might be lost to drift and forgetting may occur
- *Disengagement* remedies have traditionally kept suboptimal individuals in the population
  - Empirically, greedy algorithms which consolidate around present best tend to disengage
  - Suboptimal individuals may provide gradient

Remedies

Cycling

Ways to address cycling include:

- Fitness sharing
- Memory mechanisms
- Enrich the environment
- Multiple populations

Remedies

Cycling

Bullock 1995

- "True" coevolution is direct reciprocal evolution between two populations
- "Diffuse" coevolution entails evolutionary change in response to traits in several other populations
- Diffuse coevolution leads to more robust strategies
- Follow-up by Hornby & Mirtich 1999

Remedies

Cycling

Rosin & Belew 1995

- Zero-sum games (symmetric or asymmetric)
- Competitive fitness sharing
- Score you get against an opponent is divided by sum of all scores obtained against that opponent

Remedies

Cycling

Rosin & Belew 1995

Standard fitness calculation:

	A	B	C	D	
W	1	1	1	0	1 + 1 + 1 + 0 = 3
X	1	1	0	0	1 + 1 + 0 + 0 = 2
Y	1	0	0	0	1 + 0 + 0 + 0 = 1
Z	0	0	1	1	0 + 0 + 1 + 1 = 2

Remedies

Cycling

Rosin & Belew 1995

Competitive fitness sharing:

	A	B	C	D	
W	1	1	1	0	1/3 + 1/2 + 1/2 + 0 = 4/3
X	1	1	0	0	1/3 + 1/2 + 0 + 0 = 5/6
Y	1	0	0	0	1/3 + 0 + 0 + 0 = 1/3
Z	0	0	1	1	0 + 0 + 1/2 + 1 = 3/2
	3	2	2	1	

Remedies

Cycling

Juillé & Pollack 1996

- Fitness based on unique “covering”
- Individuals in Population 1 interact with opponents in Population 2
- Fitness of an individual determined by comparing performance with other individuals in *same* population
- Points for beating opponents that others do not beat

Remedies

Cycling

Juillé & Pollack 1996

Standard fitness calculation:

	A	B	C	D	
W	1	1	1	0	1 + 1 + 1 + 0 = 3
X	1	1	0	0	1 + 1 + 0 + 0 = 2
Y	1	0	0	0	1 + 0 + 0 + 0 = 1
Z	0	0	1	1	0 + 0 + 1 + 1 = 2

Remedies

Cycling

Juillé & Pollack 1996

Covering calculation:

	A	B	C	D
W	1	1	1	0
X	1	1	0	0
Y	1	0	0	0
Z	0	0	1	1

	W	X	Y	Z
W	0	1	2	2
X	0	0	1	2
Y	0	0	0	1
Z	1	2	2	0

Remedies

Cycling

Juillé & Pollack 1996

Covering calculation:

	W	X	Y	Z
W	0	1	2	2
X	0	0	1	2
Y	0	0	0	1
Z	1	2	2	0

0 + 1 + 2 + 2 = 5

0 + 0 + 1 + 2 = 3

0 + 0 + 0 + 1 = 1

1 + 2 + 2 + 0 = 5

Cycling: Equilibria & Dynamics

- Rosin & Belew 1997 prove that any fitness equilibrium without fitness sharing is also a fitness equilibrium with fitness sharing (in zero-sum game)
- Juillé & Pollack 1996 show that their “covering” method can lead to stable *polymorphisms*

Remedies

Cycling

Nolfi & Floreano 1998

- Robotic pursuit and evasion
- Observe cyclic dynamics
- Hypothesize that a more complex environment may dampen cyclic dynamic
- Added obstacles and walls
- Found to provide significant performance boost in some runs
- On average, though, delays onset of cycling

Remedies

Cycling

Bullock, 1995 (revisited)

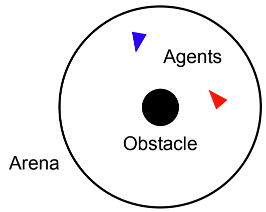
- “True” coevolution is direct reciprocal evolution between two populations
- “Diffuse” coevolution entails evolutionary change in response to traits in several other populations
- Diffuse coevolution leads to more robust strategies
- Follow-up by Hornby & Mirtich 1999

Remedies

Cycling

Hornby & Mirtich 1999

- Virtual pursuit and evasion with simulated physics of wheeled car-like agents
- Round arena with large obstacle in center



Remedies

Cycling

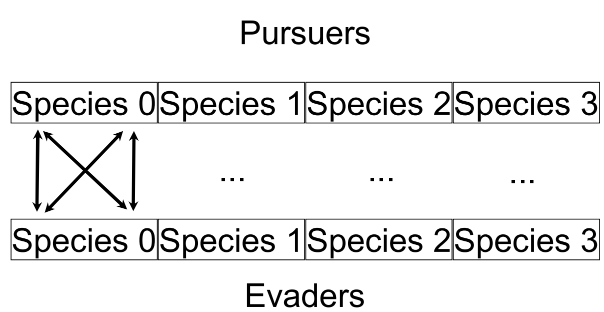
Hornby & Mirtich 1999

- Use multiple populations for each role of the game (c.f. Bullock 1995)
- Pursuers and evaders obtained under “diffuse” coevolution were more effective than those obtained from “direct” coevolution

Remedies

Cycling

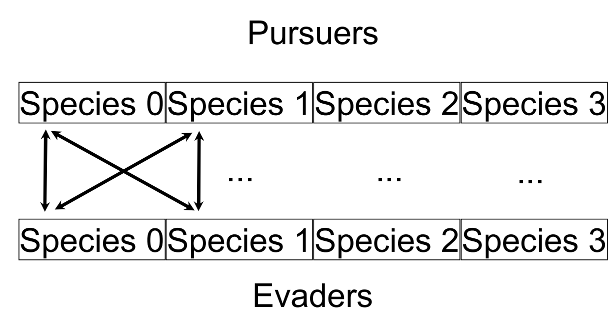
Hornby & Mirtich 1999



Remedies

Cycling

Hornby & Mirtich 1999



Remedies

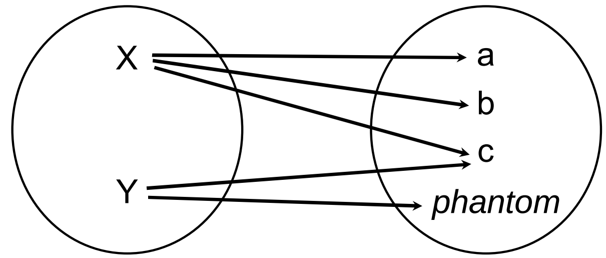
Cycling
Hornby & Mirtich 1999

- Runs using direct coevolution exhibit cyclic behavior and disengagement
- Runs using diffuse coevolution stay close to 50% wins for pursuers and evaders

Remedies

Disengagement
Rosin & Belew 1997

- "Phantom parasite" used with competitive fitness sharing to handle disengagement



Remedies

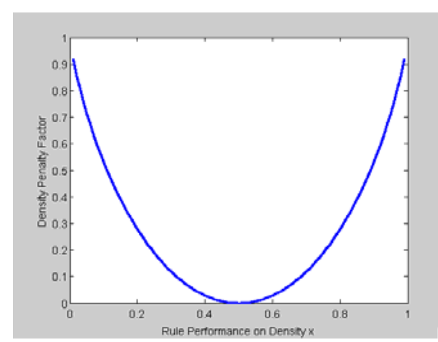
Disengagement
Juillé & Pollack 1998

- Density classification task in CA
- Purely competitive evaluation ⇒ cycling
- Competitive fitness sharing ⇒ disengage
- Penalize initial conditions with densities that cause rules to perform near random
- Should be applicable to other domains, e.g., sorting networks

Remedies

Disengagement
Rosin & Belew 1997

$$f(\mathrm{IC}_j) = \sum_{i=1}^{n_R} W(R_i^j) \times E(R_i, \rho(\mathrm{IC}_j)) \times \overline{\mathrm{covered}(R_i, \mathrm{IC}_j)}$$



Remedies

Disengagement

Olsson 1998

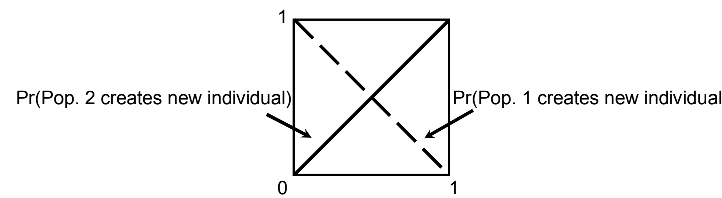
- Asymmetric zero-sum games
- Evolve only one population, leaving the other population fixed
- Evolve Pop. 1 until individual found that beats all individuals in Pop. 2
- Then evolve Pop. 2 until individual found that beats all individuals in Pop. 1

Remedies

Disengagement

Paredis 1999

- Asymmetric zero-sum games
- Steady-state algorithm
- "X-method" to decide which population gets a new individual



Remedies

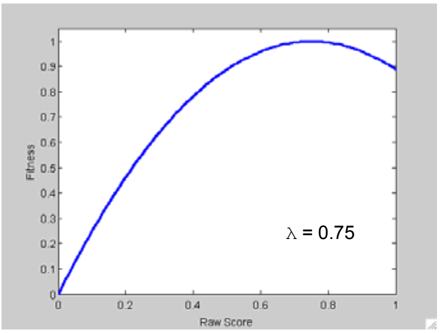
Disengagement

Cartlidge & Bullock 2002

- Moderating "parasite virulence"
- Non-monotonic function of performance

$$f(x,\lambda) = \frac{2x}{\lambda} - \frac{x^2}{\lambda^2}$$

f' indicates peak fitness at λ



Remedies

Forgetting

Boyd 1989

- Studies IPD where players can make mistakes
- Tit-For-Tat enters mutual retaliation
- Contrite Tit-For-Tat is resistant to invasion
- All-Cooperate cannot invade via drift
- Noise distinguishes CTFT from All-C

Remedies

Forgetting

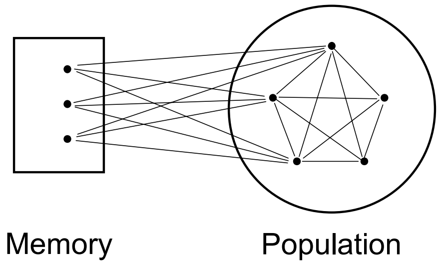
Pollack & Blair 1998

- Backgammon naturally resists forgetting
- All aspects of skill are continuously needed
- A simple hill-climber is thus able to achieve fairly impressive performance
- Estimated to achieve skill comparable to TD-Gammon rev. 1992

Remedies

Forgetting: Memory Mechanisms

- Augment evaluation by interacting with individuals stored in the memory



Remedies

Forgetting: Memory Mechanisms

- Best-of-Generation (BOG) methods
- Most-fit individual from the *m* most recent generations retained in memory
- Sample *n* of the *m* individuals with replacement to augment evaluation of population

Remedies

Forgetting: Memory Mechanisms

- Sims 1994, Cliff & Miller 1995: m = 1, n = 1
- Potter & De Jong 1994: m = 1, n = 1
- Rosin & Belew 1997: m = ∞, n = 25 & 50
- Nolfi & Floreano 1998a: m = 10, n = 10

Remedies

Forgetting: Memory Mechanisms

- BOG memory is shown to help
- Broadens selection pressure
- Stabilize algorithm behavior
- Alleviate forgetting

Monitoring

Monitoring

- The relative / subjective nature of internal fitness evaluation makes diagnosis difficult
- There are many potential pathologies
- So: A variety of methods for monitoring progress in coevolutionary systems have been developed

Monitoring

Best Elite Opponent

Sims 1994

- An early idea in monitoring
  - Both a competition patterns and a monitoring technique
  - individuals of population 1 all compete against most-fit of previous generation from population 2 (best elite)
- To monitor progress: track the outcome of each generation's competitions.
- Can indicate whether the competition is continuing

Monitoring

Hall of Fame

Rosin & Belew 1997

- "To ensure progress, we may want to save individuals for an arbitrarily long time and continue testing against them."
- *Hall of Fame*
  - stores best of each generation
  - new individuals tested against sample of Hall of Fame members
- While used as a memory mechanism, it can also function as a monitor: Track performance of new individuals by testing against the members of the hall of fame.

Monitoring

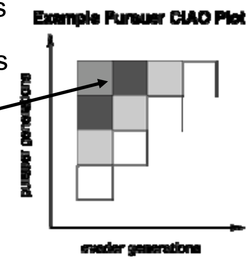
CIAO Plots
Cliff & Miller 1995

- *Current Individual vs. Ancestral Opponents* (CIAO)
- Pursuers chase evaders in a 2-D world
- Two-population coevolutionary algorithm
- “We use the term *fitness ambiguities* to refer to such cases where qualitative trends in time-series of instantaneous fitness measures could feasibly be interpreted as either continuing progress or as a breakdown of the co-evolutionary process.”

Monitoring

CIAO Plots
Cliff & Miller 1995

- Current elites play elite opponents from all previous generations
- Display outcomes in a bitmap image
  - e.g. darker for pursuers winning
- Used as a monitor of progress: if progress has occurred, present elites should be able to defeat elite opponents from previous generations



Doing well against
the previous generations

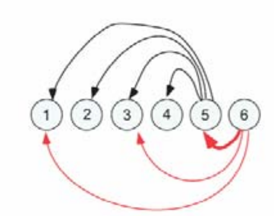
Monitoring

Master Tournament
Floreano & Nolfi 1997

- Originally demonstrated in predator/prey domain
- Adds more information to CIAO:
  - *All* best predators compete against *all* best prey.
  - Shows at which generation the overall best of each population occurred
  - Shows at which generation the most ‘interesting’ tournaments occur

Monitoring

Master Tournament Expense

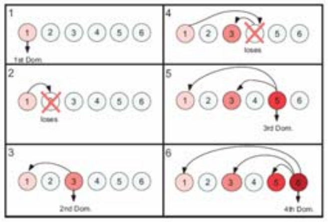


- Comparing all elite requires n^2 evaluations
  - An accurate evaluation may involve many games
- Defeating more champions does not establish unequivocal superiority

Monitoring

Faster: Dominance Tournament

Stanley & Miikkulainen 2002



Recursive Requirement prohibits circularities:

- The first dominant strategy d_1 is the generation champion of the first generation;
- dominant strategy d_j, where $j > 1$, is a generation champion such that for all $i < j$, d_j is superior to (wins the 288 game comparison with) d_i.

Monitoring

Current Population vs. Ancestral Opponent

Bader-Natal & Pollack 2005

- Compares *all of population* for each generation
  - Significantly more expensive computation
  - Significantly more information in output
  - However, can be constrained to a subset of total history to save resources
- Extended to *population differential analysis* that displayed failure conditions in addition to successes

Tutorial Summary

- Provided background and introduction to coevolutionary algorithms
- Outlined early work and notable results
- Discussed work on pathological algorithm behavior and proposed remedies
- Raised the question: what do we really want coevolution to do?

What Do We Want Coevolutionary Algorithms To Be Doing?

- **Creating Arms Races** (Ficici & Pollack 1998). "The key to successful coevolutionary learning is a *competitive arms race* between opposed participants."
- **Optimizing Robustness** (Wiegand & Potter 2006). "CCEAs…are adaptive optimizers of robustness."

What Do We Want Coevolutionary Algorithms To Be Doing?

- **Complexifying** (Stanley & Miikkulainen 2004). “Complexification encourages continuing innovation by elaborating on existing solutions.”
- **Implementing Solution Concepts** (Ficici 2004). “We assert that pathologies in coevolutionary optimization arise when algorithms fail to implement the required (or desired) solution concepts.”

Looking Forward

Looking Forward

- Solution Concepts
  - Addresses question of what a coevolutionary algorithm should output
- Pareto Coevolution
  - Treats evaluational issues
- Compositional Coevolution and Robustness
  - Treats composing evolved subparts into wholes
- NEAT and Complexification
  - Treats issues of representation

Looking Forward

Solution Concepts

- Formally specifies which individuals are part of solutions
- Fundamental questions:
  - Are common/intuitive notions of solution reasonable?
  - What solution concepts do we know, and how can we find new ones?
  - Given a solution concept, how do we know if an algorithm actually approximates it?

Looking Forward

Pareto Coevolution

- Focuses on discriminating among and evaluating candidate solutions.
- Fundamental questions:
  - Which individuals are “good,” and why?
  - How do we turn the Pareto Optimal Set into a working solution?
  - How can we deal with the “curse of dimensionality”?
  - Are memory or archive mechanisms necessary?

Looking Forward

Cooperative/Compositional Coevolution

- Evolving populations of components that can be assembled into capable composite solutions
- Fundamental questions:
  - What makes a good component?
  - What makes a good composite?
  - What kind of design choices help CCEAs find optimal composite solutions?
  - Are CCEAs naturally suited for producing composite solutions “robust” to changes in some of the components?

Looking Forward

NEAT and Complexification

- Focuses on representing complicated objects in open-ended domains.
- Fundamental questions:
  - Can we remedy pathologies by elaborating on/complexifying present solutions, versus simply altering them?
  - Can continuous, open-ended progress be achieved?

References

Angeline, P. J. and Pollack, J. B. (1993). Competitive environments evolve better solutions for complex In the 5th International Conference on Genetic Algorithms.

Axelrod, R. (1987). The evolution of strategies in the iterated Prisoner's Dilemma. In Genetic Algorithms and Simulated Annealing.

Bader-Natal, A. and Pollack, J. (2005). Towards Metrics and Visualizations Sensitive to Coevolutionary Failures. In Workshop for the 2005 AAAI Fall Symposium on Coevolutionary and Coadaptive Systems.

Barricelli, N. A. (1963). Numerical Testing of Evolution Theories. Part 11. Preliminary Tests of Performance. Symbiogenesis and Terrestrial Life. Acta Biotheoretica 16(3/4).

Boyd, R. (1989) Mistakes Allow Evolutionary Stability in the Repeated Prisoner's Dilemma Game. Journal of Theoretical Biology 136.

Bullock, S. (1995) Co-evolutionary design: Implications for evolutionary robotics. Technical Report - CSRP 384 , University of Sussex, United Kingdom.

Cartlidge, J., & Bullock, S. (2002). Learning lessons from the common cold: How reducing parasite virulence improves coevolutionary optimization. In the 2002 Congress on Evolutionary Computation.

Chellapilla K and Fogel DB (1999) "Evolving Neural Networks to Play Checkers without Expert Knowledge," IEEE Trans. Neural Networks, Vol. 10:6.

References

Chellapilla K and Fogel DB (2000) “Review of Efforts to Evolve Strategies to Play Checkers as Well as Human Experts,” Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation III.

Cliff, D. and Miller, G. F. (1995). Tracking the Red Queen: Measurements of adaptive progress in co-evolutionary simulations. In the Third European Conference on Artificial Life: Advances in Artificial Life.

Ficici, S. and Pollack, J. (1998). Challenges in Coevolutionary Learning: Arms-Race Dynamics, Open-Endedness, and Mediocre Stable States. In the Sixth International Conference on Artificial Life.

Ficici, S. (2004). Solution Concepts in Coevolutionary Algorithms. Brandeis University, Ph.D, Computer.

D. Floreano and S. Nolfi. (1997). Adaptive behavior in competing co-evolving species. Mantra technical report, LAMI, Swiss Federal Institute of Technology.

Hillis, D. (1991). Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure. In Artificial Life II, SFI Studies in the Sciences of Complexity, vol. X.

Hornby, G. S. and Mirtich, B. (1999). Diffuse versus True Coevolution in a Physics-based World. In the 1999 Genetic and Evolutionary Computation Conference.

Juillé, H. and Pollack, J. (1996). Dynamics of co-evolutionary learning. In From Animals to Animats 4. In the Fourth International Conference on Simulation of Adaptive Behavior.

References

Juillé, H. and Pollack, J. (1998). Coevolutionary Learning: a Case Study. In the Fifteenth International Conference on Machine Learning

Mitchell, M., Thomure, M. D., and Williams, N. L. (2006). The role of space in the success of coevolutionary learning. In Artificial Life X: the Tenth International Conference on the Simulation and Synthesis of Living Systems.

Nolfi, S., and Floreano, D. (1998) Co-evolving predator and prey robots: Do 'arms races' arise in artificial evolution? Artificial Life, vol. 1(1).

Olsson, B., (1998), A Host-Parasite Genetic Algorithm for Asymmetric Tasks. In the 1998 European Conference on Machine Learning.

Paredis, J., (1999). Coevolutionary Algorithms. In The Handbook of Evolutionary Computation, 1st supplement.

Reed, J., Toombs, R. and Barricelli, N. A. (1967). Simulation of self-reproducing numeric patterns by data processing machines, effects of hereditary control, mutation type and crossing. Journal of Theoretical Biology 17.

Rosin, C, D., Belew, R., K., (1995). Methods for Competitive Co-evolution; Finding Opponents Worth Beating. In the Sixth International Conference on Genetic Algorithms.

Rosin, C, D., (1997), Coevolutionary Search Among Adversaries, PhD Thesis, University of California, San Diego.

References

Rosin, C. D. and Belew, R. K. (1997). New methods for competitive coevolution. Evolutionary Computation, 5(1).

Samuel, A. (1959). Some studies in machine learning using the game of checkers. IBM J. Res. Devel.

Samuel, A. (1967). Some studies in machine learning using the game of checkers: recent progress. IBM J. Res. Devel. 11.

Sims, K (1994). Evolving 3D Morphology and Behavior by Competition. In Artificial Life.

Stanley, K. O. and Miikkulainen, R. (2002). The dominance tournament method of monitoring progress in coevolution. In the Bird of a Feather Workshops, Genetic Evolutionary Computation Conference.

Stanley, K. O. and Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. Journal of Artificial Intelligence Research, 21.

Leigh Van Valen (1973). A new evolutionary law. Evolutionary Theory, 1.

R. Paul Wiegand and Mitch A. Potter (2006). Robustness in Cooperative Coevolution. In the 2006 Genetic and Evolutionary Computation Conference.

R. Paul Wiegand and Jayshree Sarma (2004). Spatial Embedding and Loss of Gradient in Cooperative Coevolutionary Algorithms. In the 8th Parallel Problem Solving from Nature.