Coevolution Tutorial

CEC 2005

Elena Popovici, R. Paul Wiegand

George Mason University, Naval Research Laboratory

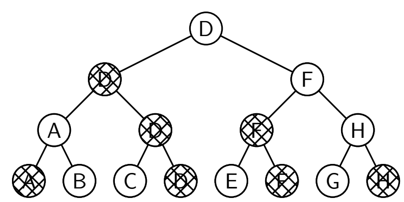
Outline

Definitions

- Coevolutionary Algorithm (CoEA) - an evolutionary algorithm (EA) in which evaluation is based on interactions between individuals.
  - Changes in the set of individuals used for evaluation can affect the ranking of individuals.
- Coevolutionary Computation (CoEC) - the subfield of computer science that is concerned with the study of CoEAs and their application to problem solving (predominantly) and modeling.
  - Encompasses both the algorithms and the problems they are applied to.

Examples

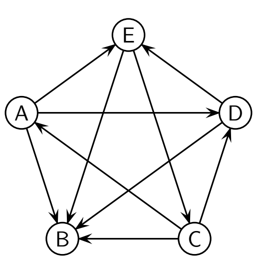
One population, single-elimination tournament



Individual	Fitness
D	4
F	3
A, H	2
B, C, E, G	1

Examples

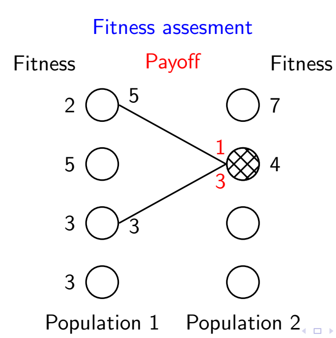
One population, full mixing



Individual	Fitness
A	1
B	4
C	1
D	2
D	2

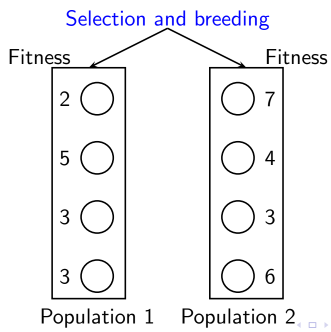
Examples

Two populations, random interacting individuals



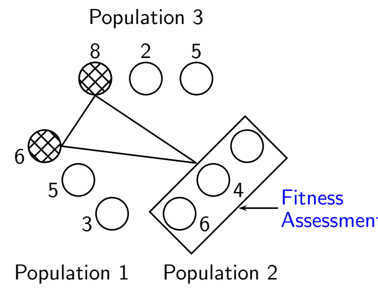
Examples

Two populations, random interacting individuals



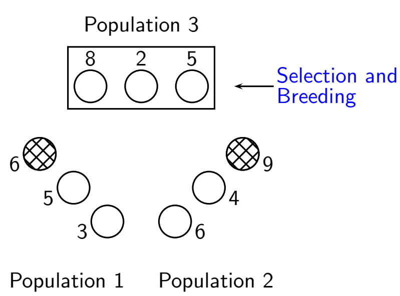
Examples

Three populations, cooperative



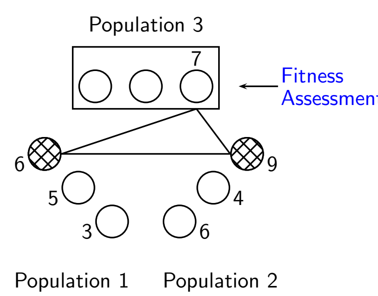
Examples

Three populations, cooperative



Examples

Three populations, cooperative



Motivation

- Simulation of processes from nature
- Problem decomposition for more efficient problem solving

Motivation

- Tackling domains in which performance of a potential solution can only be expressed by interaction with other potential solutions
  - A fitness function not based on interactions is extremely difficult / impossible to construct
  - Using the complete set of interactants is impossible
  - Using a large set of interactants is computationally expensive
  - Using a small fixed set of interactants is prone to overfitting or may not provide gradient
  - Using a small random set generally yields poor results due to sampling error

Motivation

- Domains where CoEAs may improve performance by:
  - Providing a dynamic gradient
  - Helping maintain diversity
  - Incrementally building complexity

Challenges

- Difficult to setup in order to obtain desired results
- Complex and sometimes unintuitive behaviors

Solution Concept

Solution Concept

A criterion specifying which locations in the search space are solutions and which are not (Ficici 2004).

- Engineers have a solution concept in mind when they apply an algorithm to a problem
- Algorithms *induce* a solution concept, by intention or not
  - Dynamically speaking, algorithms tend to be drawn to, or focus on, particular areas of a search space
- Key is to match algorithm's solution concept to the intended solution concept

★ Many coevolutionary algorithms fail because these are mismatched

Solution Concepts in Coevolution

Question:

What solution concept is implemented by traditional CoEAs?

- It is especially important in coevolution to have a well-defined notion of solution concept, but often there is none
- Recently, there have been several useful & well-defined coevolutionary solution concepts published:
  - Ideal Collaboration/Partnership (Wiegand 2004)
  - Maximum average/expected/cumulative payoff (Ficici 2005; de Jong 2005)
  - Nash equilibrium (Ficici & Pollack 2003)
  - Pareto-optimal set (de Jong & Pollack 2004)

Properties of Coevolutionary Problems

Question:

Are problems *inherently* coevolutionary?

Some problem properties lend themselves more toward coevolutionary systems than others:

- Underlying objectives
- Game-theoretic rewards
- Roles of problem elements

Underlying Objectives

Underlying Objective

An indicator of quality returning an element from an ordered set of scalar values, such as a real number. For any [coevolutionary] problem, a set of underlying objectives exists such that knowledge of the objective values of [a candidate provides as sufficient search information as] the outcomes of all possible [interactions] (de Jong 2004).

- Many coevolutionary problems have multiple underlying objectives
- CoEAs typically do not have direct access to these, but must infer them by testing different interactions
- Watson & Pollack (2001) describe a simple substrate for demonstrating problems with different underlying objectives
- Bucci *et al.* (2004) describe an algorithm that approximates a minimal set of underlying objectives

Game-Theoretic Rewards

One view of coevolutionary problems is one in which game-theoretic interactions are encoded directly into the problem

- Goal is to learn good players of some game
- Reward function of game is given as part of the problem
- Reward function may have various properties, e.g.:
  - Constant-sum / Variable-sum
  - Symmetry of reward
  - Other characteristics of player payoffs

Roles of Problem Elements

In coevolution, the role of elements of the problem may differ depending on the solution concept

- Object of learning/optimization may differ
  - Solving the problem may involve learning a variety of things
  - Important to understand which problem elements constitute candidate solutions and which serve supporting roles
- Different roles may be:

  Symmetric — Learn players for a game
  Asymmetric — Optimize sorters with challenging data

Categorizing Coevolutionary Problems

There are many ways to categorize coevolutionary problems based on their properties. For example:

Game-Theoretic Rewards: Use GT properties to classify problems based on the payoff from individual interactions

Roles of Problem Elements: Use the role of the solution to classify problems as *test-based* or *compositional*

Payoff from Individual Interactions

Competitive Interactions:

- *Competitor*: one of two or more striving to reach or obtain something that only one can possess (`www.m-w.com`)
- Example: constant-sum game (sufficient but not necessary)
- Can be approached with either one- or two-population algorithms
- More than two populations raises interesting denotational issues (e.g., "The enemy of my enemy is my friend.")

Competitive **problem** payoff, not competitive **algorithm**, because:

- Fitness is a function of payoffs from multiple interactions!
- Increase in (best/average) fitness values in one population **may** or **may not** cause decrease in fitness values in the opposite population

Payoff from Individual Interactions

Cooperative Interactions:

- To *cooperate*: to associate or act together with another or others for mutual benefit (`www.m-w.com`)
- Example: variable-sum game with symmetric payoff (sufficient but not necessary)
- Often useful problem decompositions lead to cooperative interactions among components

Payoff from Individual Interactions

Mixed Interactions:

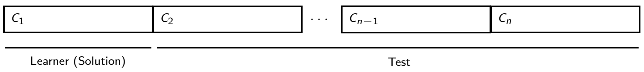
- Interactions can be a mixture of both competitive and cooperative rewards
- The majority of games are of this type
- Example: iterated prisoner's dilemma (Axelrod 1989)

Summary

- Constant-sum games are competitive
- Variable-sum games with symmetric payoff are cooperative
- Variable-sum games with asymmetric payoff can be any

Test-Based vs. Compositional

Let $C_1, C_2, ..., C_n$ be the n classes of elements involved in a problem specification



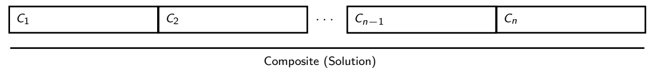
Test-based — *Solutions* specify elements from a single class

- Example: Sorting networks and challenging data
- Recall: Object of learning may differ
  - Some problem elements constitute candidate solutions and some serve supporting roles
  - Here, one class designated as the *learner* (or *candidate*), other class(es) designated as *tests* for the learner
  - For some test-based problems, roles can change during learning

Historically, called "competitive coevolution"

Test-Based vs. Compositional

Let $C_1, C_2, ..., C_n$ be the n classes of elements involved in a problem specification



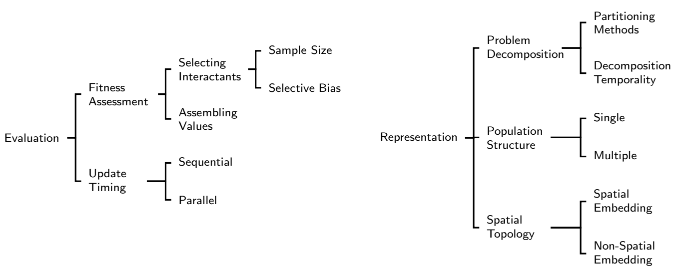
Compositional — *Solution* composite of elements from all classes

- Example: Team of cooperating agents
- Here, all elements of the problem serve a direct role in the assembled solution

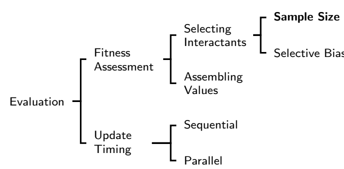
Historically, called "cooperative coevolution"

Mixed — *Solutions* specify elements from a subset of classes

Categorizing CoEAs



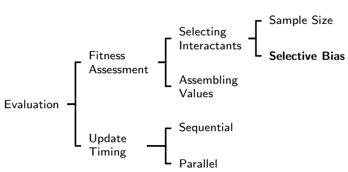
Categorizing CoEAs



- Same for all individuals
  - One
  - Few
  - All — Full mixing
- Variable
  - Single-elimination tournament

⋆ Popovici & De Jong 2005 AAAIFS; Panait & Luke 2002; Wiegand *et al.* 2001; Bull 2001; Bull 1997; Angeline & Pollack 1993

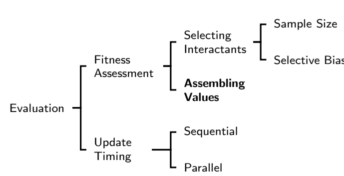
Categorizing CoEAs



- Random
- Fitness Based, e.g.:
  - *single-best collaboration* (Potter 1997)
  - *last elite opponent* (Sims 1994)
- Mixed
- Neighborhood (spatial embedding)
- Memory based, e.g.:
  - Hall of fame (Rosin & Belew 1997)

⋆ Popovici & De Jong 2005 AAAIFS; Wiegand *et al.* 2001

Categorizing CoEAs



- Single value
  - Best (mainly in traditional compositional settings)
  - Average (mainly in traditional test-based settings)
  - Competitive fitness sharing (Rosin & Belew 1995)
- n-uplu
  - Requires multi-objective-like selection methods (Ficici & Pollack 2001)

Categorizing CoEAs

- Evaluation
  - Fitness Assessment
    - Selecting Interactants
      - Sample Size
      - Selective Bias
    - Assembling Values
  - **Update Timing**
    - Sequential
    - Parallel

- Differences more complicated than might be expected; more than just separability is at stake (Jansen & Wiegand 2004)

Categorizing CoEAs

- Problem separability (Jansen & Wiegand 2004)
- Inter-population epistasis (Wiegand 2004; Bull 2001)
- Decompositional bias: matching decomposition of the representation to problem separability (Wiegand *et al.*)

- Representation
  - Problem Decomposition
    - **Partitioning Methods**
    - Decomposition Temporality
  - Population Structure
    - Single
    - Multiple
  - Spatial Topology
    - Spatial Embedding
    - Non-Spatial Embedding

Categorizing CoEAs

- Static (Potter 1997)
- Dynamic (Potter & De Jong 2000)
- Adaptive

- Representation
  - Problem Decomposition
    - Partitioning Methods
    - **Decomposition Temporality**
  - Population Structure
    - Single
    - Multiple
  - Spatial Topology
    - Spatial Embedding
    - Non-Spatial Embedding

Categorizing CoEAs

- In "symmetric" domains (i.e., with a single role) need to make a choice between one or two populations

- Representation
  - Problem Decomposition
    - Partitioning Methods
    - Decomposition Temporality
  - **Population Structure**
    - Single
    - Multiple
  - Spatial Topology
    - Spatial Embedding
    - Non-Spatial Embedding

Categorizing CoEAs

- Specialized choices about interaction methods and selection / survival
- Historically successful (Pagie1999;Cliff & Miller 1995; Hillis 1991)
- Helps maintain diversity of potential interactions (Williams & Mitchell 2005; Wiegand & Sarma 2004)

- Representation
  - Problem Decomposition
    - Partitioning Methods
    - Decomposition Temporality
  - Population Structure
    - Single
    - Multiple
  - **Spatial Topology**
    - Spatial Embedding
    - Non-Spatial Embedding

Why Study Dynamics?

- Coevolution seems a natural, highly adaptive search method for certain kinds of problems.
- Unfortunately, coevolutionary algorithms often disappoint engineers with poor performance and/or counterintuitive behavior
- Even modifications of the algorithms often lead to counterintuitive response on certain problems

⋆ Analyzing the dynamics (i.e., run-time behaviors) is key to understanding coevolution

Relativity & Its Effects

- The Red Queen — a character in Lewis Carroll's "Through the Looking Glass"
  - Imported first in biology and then in CoEC and interpreted in many different ways
  - A metaphor for relativity — the main feature of co-evolution: (internal) fitness is subjective
- Relativity's hope: to have an algorithm achieve a certain goal without specifically encoding it internally into fitness
- Relativity's caveats:
  - Generates intricate run-time behaviors
  - Makes it difficult to monitor progress towards the goal

E.g.: Wiegand 2004; de Jong & Pollack 2004; Watson & Pollack 2001; Pagie & Hogeweg 2000; Cliff & Miller 1995; Ridley 1993

Instrumenting CoEAs

- To understand how CoEA behaviors, many have attempted to measure their progress
- Metrics for the quality of individuals:
  - Contextual Dependence
    - subjective — the quality of an individual depends on the context in which it is evaluated
    - objective — context independent
  - Influence on the algorithm
    - internal — its values are used by the algorithm and influence its course
    - external — not internal

Instrumenting CoEAs

- Metrics for the quality of individuals
  - Traditional EA — internal objective metric as fitness.
  - CoEA — internal subjective metric as fitness; the context is a set of other evolving individuals.
  - Performance towards the goal — should be measured with an objective metric (in co-evolution, this will be external, as the internal measure is always subjective).

(Popovici & De Jong 2005 CEC; Watson & Pollack 2001)

Instrumenting CoEAs

- Metrics for the quality of individuals (Cont')
  - Other external metrics — designed to understand how an algorithm is functioning, but do not necessarily say anything about progress towards the goal.
    - External objective metrics
      - Ficici & Pollack 1998 — order theory
    - External subjective metrics:
      - Bader-Natal & Pollack 2004 — all of gen. ancestor contests
      - Stanley & Miikkulainen 2002 — dominance tournament
      - Funes & Pollack 2000 — information theory
      - Floreano & Nolfi 1997 — master tournament
      - Cliff & Miller 1995 — CIAO
  - Monitor best individual value / whole population average
  - Observe trends: increasing, decreasing, stagnating, noisy, repeated values, etc.

Instrumenting CoEAs

- Monitoring genotypic / phenotypic change
  - Examples:
    - Elite bitmaps
    - Ancestral Hamming maps (Cliff & Miller 1995)
    - Best-of-gen space trajectories (Popovici & De Jong 2004-5)
    - Trajectories for percentage of best in population (Wiegand 2004)
    - Dynamical system cob-web plots (Ficici *et al.* 2000)
  - Monitor best individual / whole population
  - Observe trends: cycling, converging, diverging, chaotic, etc.

Arms Race

Arms Race
Any competition where there is no absolute goal, only the relative goal of staying ahead of the other competitors (Wikipedia)

- Ideally, we hope that coevolution progresses by making parallel adaptive changes in responding strategies
- Example: Sorting networks learn to sort easy data, but the data learns to be more challenging, so the sorting network must learn to sort harder data, but the data ...
- In CoEC, we hope arms race behaviors produce a continuous increase in external objective progress toward the goal
- Unfortunately, there are several pathological dynamics that prevent arms race behaviors...

Cycling

- Visiting the same areas of the space multiple times in a repeated sequence
- Traced by observing changes in individuals at the genotypic / phenotypic level
- Generates some repeating patterns in the internal / external metrics
- Presumed related to intransitivities in the problem's definition

⋆ Popovici & De Jong 2005 CEC; Watson & Pollack 2001; Nolfi & Floreano 1997; Ficici & Pollack 1998; Juillé & Pollack 1998; Paredis 1997; Cliff & Miller 1995

Mediocre Stable States

- The external metric is "trapped" in a suboptimal value or set of values
- There may still be genotypic change occurring (e.g., cyclic or chaotic)
- Suggestions to replace the term with more specific ones

⋆ Ficici & Pollack 1998

Lack (Loss) of Gradient

- The distribution of fitness values in (at least) one population is (almost) flat
- Example: tests are too tough for learners; no learner solves any test
- May or may not be accompanied / caused by lack of diversity
- May be temporary (gradient may be regained due to changes in either population) or persistent
- Can cause genetic drift

⋆ Bucci *et al.* 2004; Wiegand & Sarma 2004; Watson & Pollack 2001; Juillé & Pollack 1998

Overspecialization

- Individuals improve on some of the underlying objectives but fail to do so on others
  - May be due to lack of representatives of those latter objectives
- Example: players discovering an opponents weaknesses and exploiting them but failing to learn the task in a general way
- Hard to detect

⋆ Bucci *et al.* 2004; Watson & Pollack 2001

Relative Overgeneralization

- Components that perform well in combination with a large number of other components are favored over components that perform very well (optimally) in combination with a small number of other components but poorly otherwise
- It is more likely to occur when the fitness results from averaging payoffs from many sampled interactions
- Depending on the goal, it can be good (e.g. when robustness, good cumulative / average performance is desired) or bad (when optimal possible payoff is desired)

⋆ Wiegand 2004

Monotonicity

- A good CoEA consistently refines solutions to the problem
- Ideally, we want algorithms that make *monotonic progress* toward the goal
- Monotonic solution concepts (Ficici 2005):
  - Use the solution concept to develop a preference relation over search space, creating a partial-ordering
  - A monotonic search process produces solution estimates that never contradict the partial ordering
  - A monotonic solution concept *guarantees* such operation
  - Algorithms that implement such solution concepts monotonically approach the solution
  - Example: Nash solution concept

Monotonicity

- Monotonic CoEAs and ideal evaluation (de Jong 2004):
  - Coevolution selects the tests used in evaluation adaptively
  - Ideal evaluation is one that uses a small set of tests but is equivalent to evaluating against all tests
  - At every point in time, only the relations between current candidate solutions need to be evaluated
  - A set of tests that reveal all relations between the existing candidate solutions is suficient.
  - Example: DELPHI
- Monotonic CoEAs and archives (de Jong 2005):
  - Monotonic CoEAs achieve reliable progress by using an archive of tests that guarantee monotonic progress
  - There is a distance function for which the distance to the solution concept decreases with every change to the archive
  - Example: MaxSolve

Empirical Analysis

- Component analysis: algorithm properties + problem properties → performance
  - Methods of interaction + cross-population epistasis in cooperative setups
    Bull 2001; Wiegand *et al.* 2001; Bull 1997; Potter 1997
  - Methods of interaction + noise in competitive setups
    Panait & Luke 2002
  - Methods of interaction + decompositional bias
    Wiegand *et al.* 2003
  - Different algorithms + asymmetry
    Olsson 2001
  - Different algorithms + intransitivity
    de Jong 2004
  - Memory & diversity maintenance + cycling
    Rosin & Belew 1997

Empirical Analysis

- Dynamics analysis
  - Run-time properties in isolation
    Cliff & Miller 1995; Stanley & Miikkulainen 2002
  - Run-time properties → performance
    Pagie & Mitchell 2002; Juillé & Pollack 1998
  - Algorithm properties → run-time properties
    Ficici & Pollack 2000
  - Problem properties → run-time properties → performance
    Watson &Pollack 2001
  - Algorithm properties + problem properties → run-time properties → performance
    Popovici & De Jong 2005 CEC; Popovici & De Jong 2005 AAAIFS

Theoretical Analysis

- Asymptotic run time analysis of the algorithms
  Jansen & Wiegand 2004;
- EGT, Dynamical systems analysis, & Markov Models
  Wiegand 2004; Ficici 2004; Schmitt 2003; Liekens 2002
- Category / Order theory
  Ficici 2004; Bucci *et al.* 2004
- Multiobjective techniques & Pareto dominance
  de Jong 2004
- Random walk & intransitivity
  Funes & Pujals 2005

Conclusions

- Coevolution can be useful for certain kinds of problems
- It can also behave in complex and counterintuitive ways
- The key to successful applications of coevolution revolves around the solution concept:

Understand your problem: What do you want to achieve?
Understand your algorithm: To what types of solutions is it drawn?
Match the two: Use the right tool for the right job

Bibliography

P. J. Angeline and J. B. Pollack. Competitive environments evolve better solutions for complex tasks. In *5th International Conference on Genetic Algorithms*, pages 264–270, 1993.

R. Axelrod. Evolving new strategies. In L. Davis, editor, *Genetic Algorithms and Simulated Annealing*, 1987-89.

A. Bader-Natal and J. B. Pollack. A population-differential method of monitoring success and failure in coevolution. In *Genetic and Evolutionary Computation Conference*. Springer Verlag, 2004.

A. Bucci, J. Pollack, and E. de Jong. Automated extraction of problem structure. In *Genetic and Evolutionary Computation Conference*, volume 3102 of *Lecture Notes in Computer Science*. Springer Verlag, 2004.

L. Bull. Evolutionary computing in multi-agent environments: Partners. In T. Baeck, editor, *7th International Conference on Genetic Algorithms*. Morgan Kaufmann, 1997.

L. Bull. Evolutionary computing in multi-agent environments: Operators. In D. Wagen and A. E. Eiben, editors, *7th International Conference on Evolutionary Programming*. Springer Verlag, 1998.

L. Bull. On coevolutionary genetic algorithms. *Soft Computing*, 5(3):201–207, 2001.

D. Cliff and G. F. Miller. Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations. In *European Conference on Artificial Life*, 1995.

E. de Jong. The maxsolve algorithm for coevolution. In H.-G. B. others, editor, *Genetic and Evolutionary Computation Conference*. ACM Press, 2005.

E. de Jong and J. B. Pollack. Ideal evaluation from coevolution. *Evolutionary Computation Journal*, 12(2):159–192, 2004.

E. D. de Jong. Intransitivity in coevolution. In X. Yao et al., editors, *Parallel Problem Solving from Nature*, 2004.

S. G. Ficici. *Solutions concepts in Coevolutionary Algorithms*. PhD thesis, Brandeis University Department of Computer Science, 2004.

S. G. Ficici. Monotonic solution concepts in coevolution. In H.-G. B. others, editor, *Genetic and Evolutionary Computation Conference*. ACM Press, 2005.

S. G. Ficici, O. Melnik, and J. B. Pollack. A game-theoretic investigation of selection methods used in evolutionary algorithms. In A. Zalzala et al., editors, *International Congress on Evolutionary Computation*. IEEE Press, 2000.

S. G. Ficici and J. B. Pollack. Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states. In C. Adami et al., editors, *Artificial Life VI*. The MIT Press, 1998.

S. G. Ficici and J. B. Pollack. A game-theoretic approach to the simple coevolutionary algorithm. In M. Schoenauer et al., editors, *Parallel Problem Solving from Nature VI*. Springer Verlag, 2000.

Bibliography

S. G. Ficici and J. B. Pollack. Pareto optimality in coevolutionary learning. In P. S. J. Kelemen, editor, *Advances in Artificial Life: 6th European Conference*. Springer Verlag, 2001.

S. G. Ficici and J. B. Pollack. A game-theoretic memory mechanism for coevolution. In E. Cantu-Paz et al., editors, *Genetic and Evolutionary Computation Conference*. Springer, 2003.

D. Floreano and S. Nolfi. God save the red queen! competition in co-evolutionary robotics. In J. R. Koza et al., editors, *Genetic Programming Conference*. Morgan Kaufmann, 1997.

P. Funes and J. B. Pollack. Measuring progress in coevolutionary competition. In J. Meyer et al., editors, *From Animals to Animats 6: Sixth International Conference on Simulation of Adaptive Behavior*. MIT Press, 2000.

W. D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. In C. G. Langton et al., editors, *Artificial Life II*. Westview Press, 1990.

T. Jansen and R. P. Wiegand. The cooperative coevolutionary (1+1) ea. *Evolutionary Computation Journal*, 12(4), 2004.

H. Juillé and J. B. Pollack. Coevolving the ideal trainer: Application to the discovery of cellular automata rules. In J. R. Koza et al., editors, *Genetic Programming Conference*. Morgan Kaufmann, 1998.

A. Liekens. *Evolution of Small Populations in Dynamic Environments*. PhD thesis, Technische Universiteit Eindhoven, the Netherlands, 2005.

B. Olsson. Co-evolutionary search in asymmetric spaces. *Information Sciences*, 133(3-4):103–125, 2001.

L. Pagie. *Information integration in evolutionary processes*. PhD thesis, Proefschrift Universiteit Utrecht, 1999.

L. Pagie and P. Hogeweg. Information integration and red queen dynamics in coevolutionary optimization. In *Congress on Evolutionary Computation*. IEEE, 2000.

L. Pagie and M. Mitchell. A comparison of evolutionary and coevolutionary search. *International Journal of Computational Intelligence and Applications*, 2(1):53–69, 2002.

L. Panait and S. Luke. A comparison of two competitive fitness functions. In W. B. Langdon et al., editors, *Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 2002.

J. Paredis. Coevolving cellular automata: Be aware of the red queen. In T. Bäck, editor, *7th International Conference on Genetic Algorithms*. Morgan Kaufmann, 1997.

E. Popovici and K. De Jong. Understanding competitive coevolutionary dynamics via fitness landscapes. In S. Luke, editor, *AAAI Fall Symposium on Artificial Multiagent Learning*. AAAI Press, 2004.

E. Popovici and K. De Jong. A dynamical systems analysis of collaboration methods in cooperative co-evolution. In *AAAI Fall Symposium Series Co-evolution Workshop*, 2005. to appear.

Bibliography

E. Popovici and K. De Jong. Relationships between internal and external metrics in co-evolution. In *Congress on Evolutionary Computation*, 2005. to appear.

E. Popovici and K. De Jong. Understanding cooperative coevolutionary dynamics via simple fitness landscapes. In H.-G. Beyer et al., editors, *Genetic and Evolutionary Computation Conference*. ACM Press, 2005.

M. Potter. *The Design and Analysis of a Computational Model of Cooperative Coevolution*. PhD thesis, George Mason University, Computer Science Department, 1997.

M. Potter and K. De Jong. Cooperative coevolution:an architecture for evolving coadapted subcomponents. *Evolutionary Computation Journal*, 8(1):1–29, 2000.

M. Ridley. *The Red Queen*. Macmillan Pub Co, 1993.

C. D. Rosin and R. K. Belew. Methods for competitive coevolution: Finding opponents worth beating. In *6th International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., 1995.

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

L. Schmitt. Coevolutionary convergence to global optima. Technical report, The University of Aizu, Aizu-Wakamatisu City, Japan, 2003.

K. Sims. Evolving 3D morphology and behaviour by competition. In R. Brooks and P. Maes, editors, *Artificial Life IV Proceedings*. MIT Press, 1994.

K. O. Stanley and R. Miikkulainen. The dominance tournament method of monitoring progress in coevolution. In W. B. Langdon et al., editors, *Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 2002.

R. A. Watson and J. B. Pollack. Coevolutionary dynamics in a minimal substrate. In L. Spector et al., editors, *Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 2001.

R. P. Wiegand. *An Analysis of Cooperative Coevolutionary Algorithms*. PhD thesis, George Mason University, Fairfax, VA, 2004.

R. P. Wiegand, W. Liles, and K. De Jong. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In L. Spector, editor, *Genetic and Evolutionary Conference*. Morgan Kaufmann, 2001. Errata available at http://www.tesseract.org/paul/papers/gecco01-cca-errata.pdf.

R. P. Wiegand and J. Sarma. Spatial embedding and loss of gradient in cooperative coevolutionary algorithms. In X. Yao et al., editors, *Parallel Problem Solving from Nature*. Springer, 2004.

N. Williams and M. Mitchell. Investigating the success of spatial coevolution. In *Genetic and Evolutionary Computation Conference*, 2005.

Questions & Comments

Elena Popovici
epopovic@gmu.edu

EClab
George Mason University
http://cs.gmu.edu/∼epopovic

R. Paul Wiegand
paul@tesseract.org

NCARAI
Naval Research Laboratory
http://www.aic.nrl.navy.mil