Evolutionary Computation: A Unified Approach

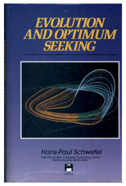
Kenneth De Jong

Computer Science Department
George Mason University
kdejong@gmu.edu
www.cs.gmu.edu/~eclab

GECCO '17 Companion, July 15-19, 2017, Berlin, Germany
ACM 978-1-4503-4939-0/17/07.
http://dx.doi.org/10.1145/3067695.3067715

1

Historical roots:

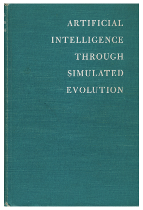


- **Evolution Strategies (ESs)**:
  - developed by Rechenberg, Schwefel, etc. in 1960s.
  - focus: real-valued parameter optimization
  - individual: vector of real-valued parameters
  - reproduction: Gaussian "mutation" of parameters
  - M parents, K>>M offspring

2

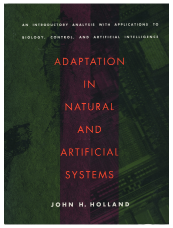
Historical roots:

- **Evolutionary Programming (EP):**
  - Developed by Fogel in 1960s
  - Goal: evolve intelligent behavior
  - Individuals: finite state machines
  - Offspring via mutation of FSMs
  - M parents, M offspring



3

Historical roots:

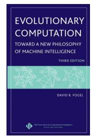


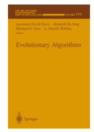
- **Genetic Algorithms (GAs)**:
  - developed by Holland in 1960s
  - goal: robust, adaptive systems
  - used an internal "genetic" encoding of points
  - reproduction via mutation and recombination of the genetic code.
  - M parents, M offspring

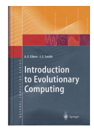
4

By the year 2000:

- A variety of evolutionary algorithms (EAs)
- A variety of applications:
  - optimization
  - search
  - learning, adaptation
- A variety of analysis tools:
  - theoretical
  - experimental



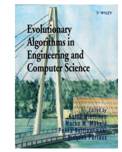


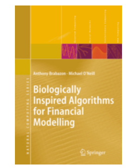


5

Interesting dilemma:

- Lots of new and difficult application opportunities.
- A bewildering variety of algorithms and approaches:
  - GAs, ESs, EP, GP, …
- Hard to see relationships, assess strengths & weaknesses, make choices, ...





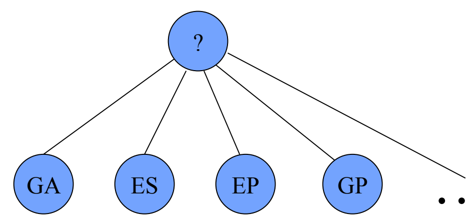
6

A Personal Interest:

- Develop a general framework that:
  - Helps one compare and contrast approaches.
  - Encourages crossbreeding.
  - Facilitates intelligent design choices.

7

Viewpoint:



8

Starting point:

- Common features
- Basic definitions and terminology

9

Common Features:

- Use of Darwinian-like evolutionary processes to solve difficult computational problems.
- Hence, the name:

Evolutionary Computation

10

Key Element: An Evolutionary Algorithm

- Based on a Darwinian notion of an evolutionary system.
- Basic elements:
  - a population of "individuals"
  - a notion of "fitness"
  - a birth/death cycle biased by fitness
  - a notion of "inheritance"

11

An EA template:

1. Randomly generate an initial population.

2. Do until some stopping criteria is met:

    Select individuals to be parents (biased by fitness).
    Produce offspring.
    Select individuals to die (biased by fitness).

    End Do.

3. Return a result.

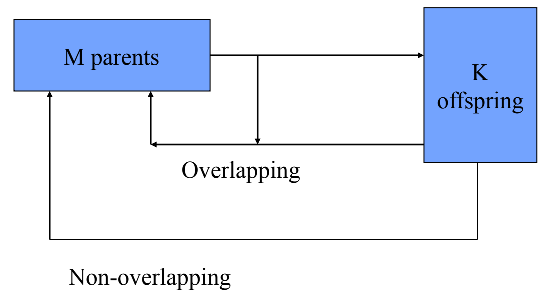
12

Instantiate by specifying:

- Population dynamics:
  - Population size
  - Parent selection
  - Reproduction and inheritance
  - Survival competition
- Representation:
  - Internal to external mapping
- Fitness

13

EA Population Dynamics:



14

Population sizing:

- Parent population size M:
  - degree of parallelism
- Offspring population size K:
  - amount of activity w/o feedback

15

Population sizing:

- Examples:
  - M=1, K small: early ESs
  - M small, K large: typical ESs
  - M moderate, K=M: traditional GAs and EP
  - M large, K small: steady state GAs
  - M = K large: traditional GP

16

Selection pressure:

- Overlapping generations:
  - more pressure than non-overlapping
- Selection strategies (decreasing pressure):
  - truncation
  - tournament and ranking
  - fitness proportional
  - uniform
- Stochastic vs. deterministic

17

Reproduction:

- Preserve useful features
- Introduce variety and novelty
- Strategies:
  - single parent: cloning + mutation
  - multi-parent: recombination + mutation
  - ...
- Price' s theorem:
  - fitness covariance

18

Exploitation/Exploration Balance:

- Selection pressure: exploitation
  - reduce scope of search
- Reproduction: exploration
  - expand scope of search
- Key issue: appropriate balance
  - e.g., strong selection + high mutation rates
  - e.g, weak selection + low mutation rates

19

Representation:

- How to represent the space to be searched?
  - **Genotypic** representations:
    - universal encodings
    - portability
    - minimal domain knowledge

20

Representation:

- How to represent the space to be searched?
  - **Phenotypic** representations:
    - problem-specific encodings
    - leverage domain knowledge
    - lack of portability

21

Fitness landscapes:

- Continuous/discrete
- Number of local/global peaks
- Ruggedness
- Constraints
- Static/dynamic

22

The Art of EC:

- Choosing problems that make sense.
- Choosing an appropriate EA:
  - reuse an existing one
  - hand-craft a new one

23

EC: Using EAs to Solve Problems

- What kinds of problems?
- What kinds of EAs?

24

Intuitive view:

- Parallel, adaptive search procedure.
- Useful global search heuristic.
- A paradigm that can be instantiated in a variety of ways.
- Can be very general or problem specific.
- Strong sense of fitness "optimization".

25

Evolutionary Optimization:

- Fitness: function to be optimized
- Individuals: points in the space
- Reproduction: generating new sample points from existing ones.

26

Useful Optimization Properties:

- Applicable to continuous, discrete, mixed optimization problems.
- No *a priori* assumptions about convexity, continuity, differentiability, etc.
- Relatively insensitive to noise.
- Easy to parallelize.

27

Real-valued Param. Optimization:

- High dimensional problems
- Highly multi-modal problems
- Problems with non-linear constraints

28

Discrete Optimization:

- TSP problems
- Boolean satisfiability problems
- Frequency assignment problems
- Job shop scheduling problems

29

Multi-objective Optimization:

- Pareto optimality problems
- A variety of difficult industrial problems

30

Properties of standard EAs:

- **GAs**:
  - universality encourages new applications
  - well-balanced for global search
  - requires mapping to internal representation

31

Properties of standard EAs:

- **ESs**:
  - well-suited for real-valued optimization.
  - built-in self-adaptation.
  - requires significant redesign for other application areas.

32

Properties of standard EAs:

- **EP**:
  - well-suited for phenotypic representations.
  - encourages domain-specific representation and operators.
  - requires significant design for each application area.

33

Other EAs:

- GP: (Koza)
  - standard GA population dynamics
  - individuals: parse trees of Lisp code
  - large population sizes
  - specialized crossover
  - minimal mutation

34

Other EAs:

- CMA-ESs: (Hansen et al)
  - Covariance Matrix Adaptation
  - ES variation to deal with parameter interactions
  - Maintains/updates matrix used to help generate useful offspring.

35

Other EAs:

- (m,k)EAs: (Wegener et al)
  - Combines ES dynamics with GA representation and operators:
    - Binary representations
    - Bit-flip mutation
  - Applied to discrete optimization problems
  - Simplicity yields strong convergence proofs

36

Other EAs:

- Differential Evolution: (Storn & Price)
  - Specifically for continuous function optimization
    - K=1 offspring
    - overlapping generations
  - parent selection: deterministic
  - 1 offspring via crossover with a 3-parent combo
  - survival selection: parent vs. offspring

37

Other EAs:

- Messy GAs (Goldberg)
- Genitor (Whitley)
- Genocop (Michalewicz)
- CHC (Eschelman et al)
- Geometric Semantic GP: (Moraglio et al)
- Gene Expression Programming (Ferreira)
- Neuroevolution (Stanley)
- …

38

Designing an EA:

- Choose an appropriate representation
  - effective building blocks
  - semantically meaningful subassemblies
- Choose effective reproductive operators
  - fitness covariance

39

Designing an EA:

- Choose appropriate selection pressure
  - local vs. global search
- Choosing a useful fitness function
  - exploitable information

Result: a well-designed EA

40

The Present …

- New applications pressing state of the art.
- Unified view of "simple EAs" is not sufficient.
- Principled extensions are required.

41

Broader Problem domains:

- Objects to be evolved:
  - Parameter values
  - Non-linear structures
  - Variable-size structures
  - Executable programs
- Goals:
  - Optimization (single, multiple objectives)
  - Adaptation (tracking, tuning)
  - Learning (induction, prediction)

42

Example: Evolving NLP Tagging Rules

- Given:
  - Existing tagging engine
  - Existing rule syntax
  - Existing rule semantics
- Goal:
  - Improve development time for new domains. by evolving tagging rule sets.
  - Improve tagging accuracy.

43

Example: Adaptive Testing

- How to validate complex systems?
  - Prove theorems?
  - Develop test suites?
  - Hire test engineers?
- Interesting alternative:
  - Use EAs to search scenario spaces.
  - Scenario's fitness related to the difficulties it creates.
    - Testing autonomous vehicle controllers

44

Example: Evolving agent behavior

- Evolve interesting/robust behavior for:
  - Web crawlers
  - Teams of robots
  - Stock market trading programs
  - War games: semi-automated forces
  - …

45

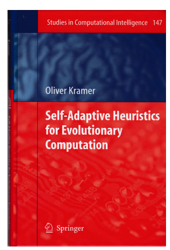
To Repeat …

- Unified view of "simple EAs" is not sufficient.
- Principled extensions are required.

46

EC extensions:

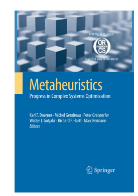
- Reduced knob twiddling:
  - To "get it right" we:
    - vary population size
    - vary selection pressure
    - vary representation
    - vary reproductive operators
  - Far better to have:
    - Principled choices
    - Self-adapting mechanisms



47

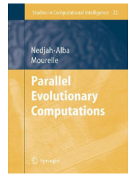
EC extensions:

- Automated EA Design:
  - Meta-heuristics
  - Hyper-heuristics
  - …

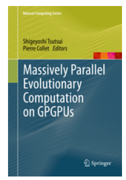


48

EC extensions:



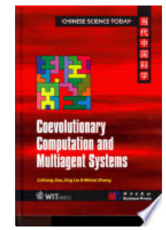
- Exploiting parallelism:
  - Low hanging fruit: parallel evaluation
  - Tougher challenges:
    - coarsely grained network models
      - isolated islands with occasional migrations
    - finely grained diffusion models
      - continuous interaction in local neighborhoods



49

EC extensions:

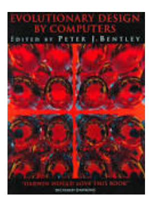
- Understanding co-evolutionary models:
  - Competitive co-evolution
    - improve performance via “arms race”
  - Cooperative co-evolution
    - evolve subcomponents in parallel
  - Agent-oriented models



50

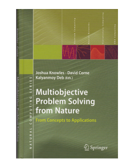
EC extensions:

- Evolutionary Design:
  - Exploring design spaces
  - Exploiting morphogenesis:
    - Sophisticated genotype --> phenotype mappings
    - Evolve plans for generating objects
  - Evolutionary art, music, …

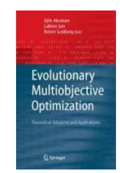


51

EC extensions:



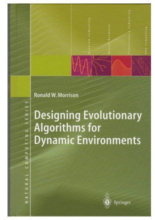
- Understand multi-objective optimization better:
  - Standard feature of industrial problems.
  - Goal: find a set of non-dominated alternatives.
  - Considerable progress already.
  - Need a deeper theoretical understanding.



52

EC extensions:

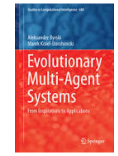
- Understand time-varying environments better:
  - Fitness landscape changes during evolution
  - Goal: adaptation, tracking
  - Considerable progress already
  - Need deeper theoretical understanding

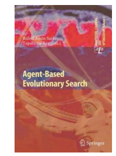


53

EC extensions:

- Agent-oriented problems:
  - Individuals more autonomous, active
  - Fitness is a function of other agents and environment-altering actions
  - E.g.,
    - Evolutionary Robotics
    - HIV evolution
    - Evolution of cooperation

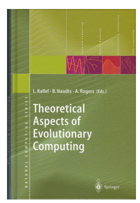


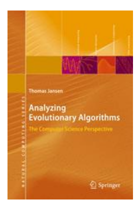


54

EC extensions:

- Need stronger analysis tools:
  - Markov models
  - Statistical mechanics
  - Evolutionary game theory
  - Test problem generators
  - Visualization

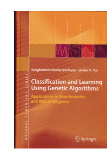




55

EC extensions:

- Need better hybrid systems:
  - Memetic algorithms: EAs and local search
  - EAs and ANNs
  - EAs and machine learning
  - EAs and agent-based models
  - …



56

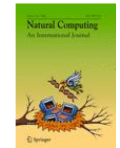
The Future …

- Continuing development of extensions
- Expanding contact with other communities:
  - Heuristic search
  - AI
  - Optimization
  - Automated design
  - …

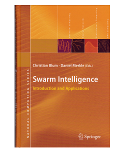
57

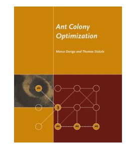
Broader picture:

"Natural Computation"



- Computational models inspired by nature:
  - Evolutionary computation
  - Simulated annealing
  - Ant colony optimization
  - Particle swarm optimization
  - Artificial neural networks
  - Artificial immune systems
  - …



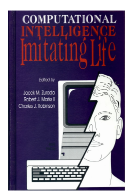


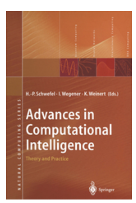
58

Broader picture:

"Computational Intelligence"

- Preferred by many over "Artificial Intelligence"
  - Evolutionary computation
  - Fuzzy systems
  - Artificial neural networks
  - …





59

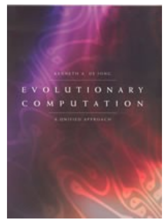
Conclusions:

- Powerful tool for your toolbox.
- Complements other techniques.
- Best viewed as a paradigm to be instantiated, guided by theory and practice.
- Success a function of particular instantiation.

60

More information:

- Journals:
  - Evolutionary Computation (MIT Press)
  - Trans. on Evolutionary Computation (IEEE)
  - Genetic Programming & Evolvable Hardware
- Conferences:
  - GECCO, CEC, PPSN, FOGA, …
- Internet:
  - www.cs.gmu.edu/~eclab
- My book:



  - Evolutionary Computation: A Unified Approach
    - MIT Press

61