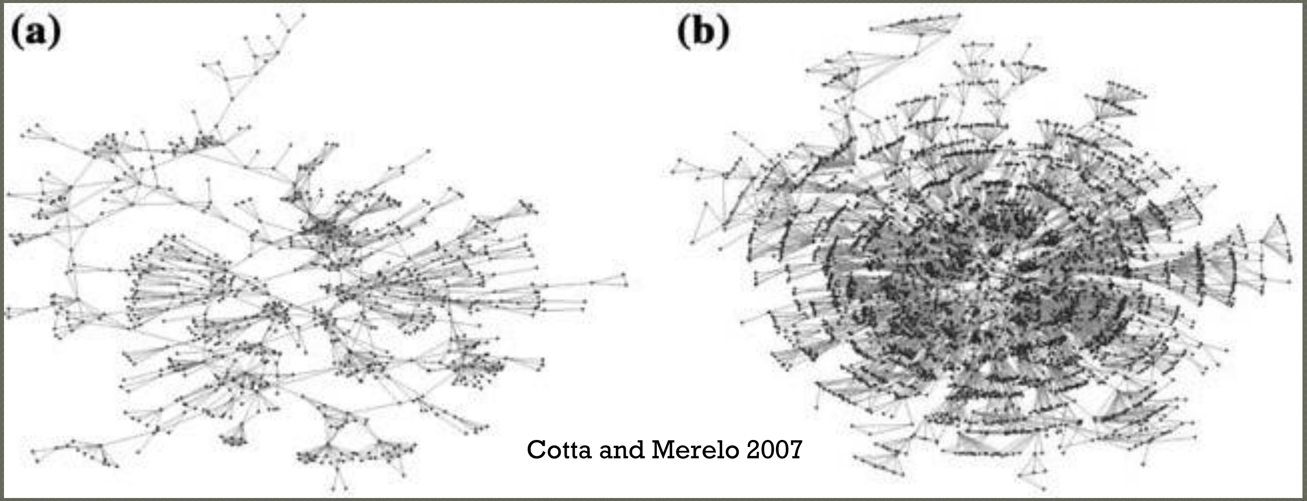
2
Heuristics and evolutionary computation

Cedric Gondro

The Centre for Genetic Analysis and Applications
University of New England

Before we get started

- Evolutionary computation (EC) is the field of study
- Evolutionary algorithm (EA) is a method
- EC is a large field of research
  - 2837 authors in 2000
  - 6555 authors in 2005 and growing



Cotta and Merelo 2007

Is it about the destination or is it about getting there?

- It’s great to be on the road, but this week we will focus on getting where we are going!
- First question: are EAs the best alternative?
- Most (technical) problems involve some form of optimization
  - Optimization is the process of searching for improved solutions in the search space

Heuristics

- From the Greek "heuriskein“ – to discover
- Pertains to the process of gaining knowledge or some desired result by intelligent guesswork rather than by following some pre-established formula
- Describes an approach to learning by trying without necessarily having an organized hypothesis or way of proving that the results proved or disproved the hypothesis
- Trial-by-error learning

When to use heuristics

- Heuristics are effective when:
  - The number of possible solutions in the search space impedes exhaustive search
  - The problem is so complicated that they demand such simplified models that any results are essentially useless
  - The evaluation function is noisy or varies with time
  - The solutions are heavily constrained

Deterministic x stochastic methods

- Deterministic
  - Always the same output for a given input
  - e.g. mathematical function
    - y=x^2
    - In computer terms, it's a description of the steps (states) to execute
  - Exhaustive search – always the same answer (almost an heuristic)

- Stochastic
  - The output cannot be predicted from the input
  - Final state does not depend on the initial state

Deterministic x stochastic - pitfalls

- Deterministic
  - Methods tend to be specific to a certain problem
  - Lack of generality
  - Most real world problems don’t come with a formula
- Stochastic
  - Slower than deterministic methods
  - Cannot prove or disprove results
  - Repeatability

Short version:
if there’s a deterministic solution to your problem, use it!

Optimization strategies
how to solve a problem

- 1 - ask the guy next door!
- 2 - the hard way. Solve it yourself!
  - Define the problem as an optimization problem
  - Define what class of problem you have
    - Assignment
    - Model parameterization
    - Model discovery
    - Combinatorial
  - Choose a method, but remember the *No-free-lunch* theorem

Complete and partial solution methods

- Complete
  - Exhaustive search
  - Gradient methods
  - Linear programming
- Partial
  - Greedy algorithms
  - Dynamic programming
  - Branch-and-bound

  - It all boils down to exploring the solution space
  - The topography of your solution spaces defines which methods are suitable
  - perfect world
  - smooth sailing
  - jagged
  - no visibility
  - welcome to Nepal!
  - bend the world to your needs!

Non EC stochastic methods

- Random search
- Random walk
- Hill climbing
- Simulated Annealing

Find the beer!

EC (at last!)

- Computer-based problem solving systems that use computational models of evolutionary processes as the key elements in design and implementation.

- Natural Computation ->

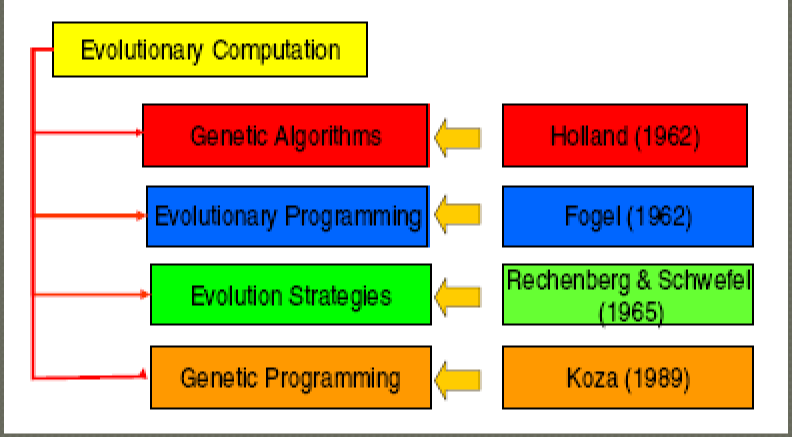
  Computational Intelligence ->

  Evolutionary Computation

- Evolutionary Computation Branches:
  - Genetic Algorithms (GAs)
  - Evolutionary Programming (EP)
  - Evolution Strategies (ES)
  - Classifier Systems (CFS)
  - Genetic Programming (GP)

Evolution of evolutionary computation

- De Jong divided EC into *epochs*
  - Catalytic 60s
  - Explorative 70s
  - Exploitative 80s
  - Unifying 90s



Evolutionary Computation

Genetic Algorithms ← Holland (1962)

Evolutionary Programming ← Fogel (1962)

Evolution Strategies ← Rechenberg & Schwefel (1965)

Genetic Programming ← Koza (1989)

Yu 2006

Types of evolutionary algorithms

- Evolutionary Programming (EP)
- Evolution Strategies (ES)
- Genetic Algorithms (GAs)
- Genetic Programming (GP)
- Classifier Systems (CFS)

… and Differential Evolution (DE)!

Which one do I use?

- NFL theorem applies!

- Rule of thumb:
  - GA – discrete problems: sorting, ranking, allocation
  - EP, ES – continuous problems: model parameterization
  - GP – model discovery

- Considerations:
  - Ease of implementation
  - Computational and convergence speeds
  - Repeatability and spread
  - Parameterization (soft-spot of EAs)

Advantages of EAs

- Simplicity
  - The concept and the implementation of EAs are simple
- Broad applicability
  - Virtually any problem can be addressed by EAs
- Hybrid methods
  - EAs allow integration with other methods
- Parallelism
  - The structure of EAs makes them particularly well suited for parallelization
- Robust to changes
  - Changes in the target system do not render the algorithm useless
- Self adaptation
  - The parameters of the EA can evolve alongside the solutions
- Solve problems with no known answers
  - Probably the greatest advantage of EAs; if an evaluation of goodness of fit of a solution is possible, EAs can be used

But EAs are not a magic bullet

- You will still have to think about the problem
  - Problem representation is critical to solving it
  - On the bright side you and the computer will teach each other!

une
University of
New England

University of New England