Meta-Heuristics

Collin Price & Ron Bond

May 12, 2014

Overview

1. Optimization
2. Metaheuristics
   - Metaheuristics? Why?
   - Classifications
3. Metaheuristic Examples
   - Genetic Algorithms
   - Particle Swarm Optimization
   - Genetic Programming
   - Solution & Fitness Representation
4. Applications
5. Metaheuristic limitations
   - No Free Lunch
   - Optimizing Optimizers

Optimization

Objective

- Minimize/maximize an objective output whose value(s) depend on a set of input variables
- Function often referred to as "Fitness Function"

Landscape

- The n-dimensional mapping of input to output values:
- inputs are parameters to be optimized
- outputs are the "fitness" or score of the given input configuration

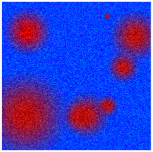


Figure: Fitness Landscape

Metaheuristics

- Partial search algorithm
- Often stochastic in nature
- Do not guarantee a globally optimal solution

Why Metaheuristics?

- Incomplete or imperfect information
- Limited computation capacity

Classifications

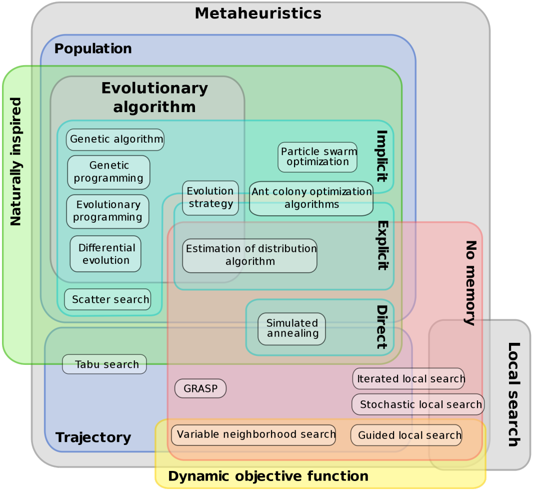
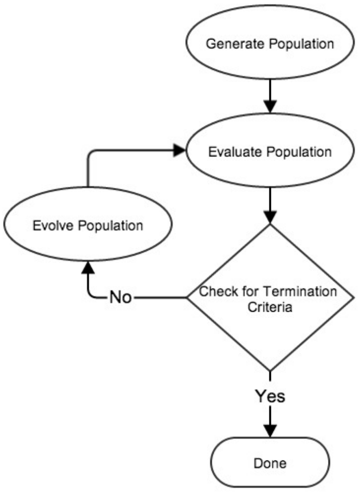


Figure: Metaheuristic Clasifications

Iterative Framework Overview

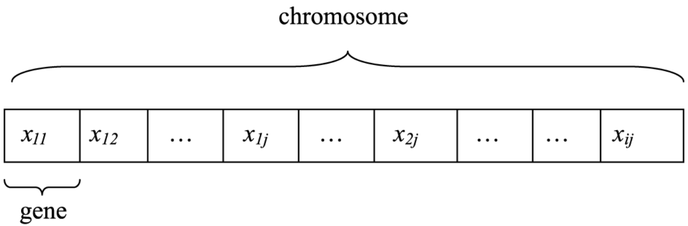


Genetic Algorithm

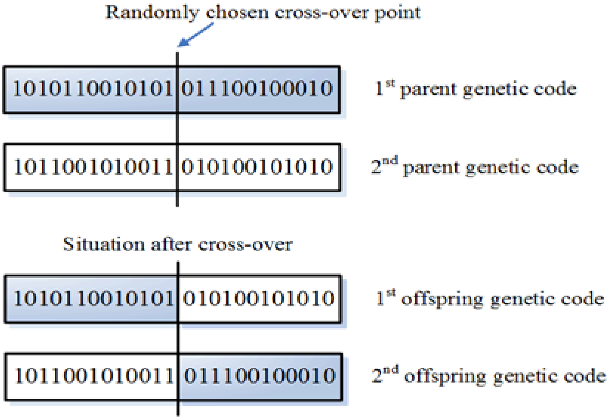
- A search heuristic that mimics the process of natural selection.
- Individuals are combined and modified to create new ones.

Individuals

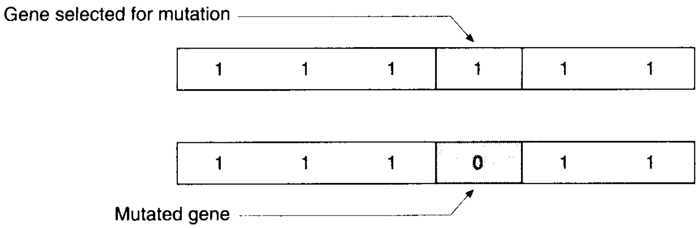
- Represents a proposed solution to the problem.
- Commonly represented as an array.
- Representations can vary depending on the problem.



Crossover


Randomly chosen cross-over point
1010110010101 | 011100100010
1st parent genetic code
1011001010011 | 010100101010
2nd parent genetic code
Situation after cross-over
1010110010101 | 010100101010
1st offspring genetic code
1011001010011 | 011100100010
2nd offspring genetic code

Mutation



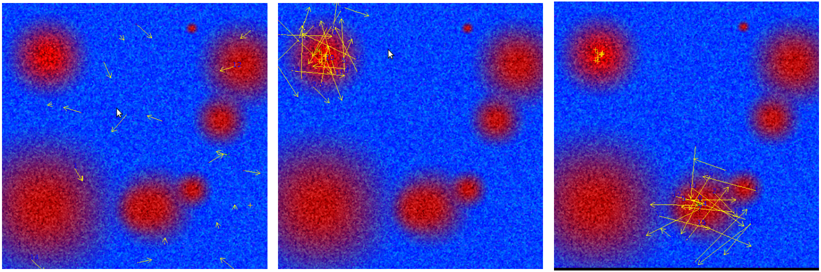
Selection

- Method used to select individuals from the population for breeding.
- Example: Tournament Selection
  - Randomly choose n-individuals from the population.
  - Select the individual with the best fitness.

Parameters

- There are several parameters that can be tuned to change the results of your genetic algorithm.
- There is no optimal set of parameters. Parameter values are empirically found.
- Parameters:
    - Number of Generations
    - Population Size
    - Crossover Rate
    - Mutation Rate
    - Elitism?

Particle Swarm Optimization



$$v_{i,d} = \underbrace{\boxed{W}}_{\text{Inertia Weight}} * v_{i,d} + \underbrace{\boxed{C_1 * Rnd(0,1) * [pb_{i,d} - x_{i,d}]}}_{\text{cognitive influence}} + \underbrace{\boxed{C_2 * Rnd(0,1) * [gb_d - x_{i,d}]}}_{\text{social influence}}$$

Figure: Original PSO Behaviour

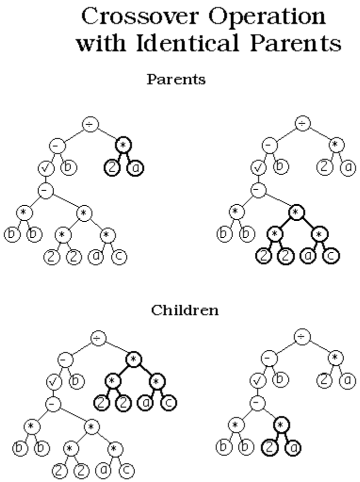
Genetic Programming

- Specialization of a genetic algorithm.
- Each individual is a computer program.
- Traditionally represented as tree structures.

Individuals

- Individuals randomly generated from a list of operators and terminals.
- Example Operators: addition, subtraction, multiplication, cosine.
- Example Terminals: ephemeral random constants, problem specific numbers.

Crossover & Mutation



Solution & Fitness Representation

- Performance of Metaheuristics are limited to the information given
- "information" is encoded in
  - Solution Representation
  - Fitness Function

Representation - Efficiency

- Search space grows with the size of the language
- Minimizing the language reduces search complexity
- Example: Language 1 (left) vs Language 2 (right)

- add(x,y)
- sub(x,y)
- mult(x,y)
- div(x,y)
- log(x)
- max(x,y)

- sub(x,y)
- mult(x,y)
- max(x,y)

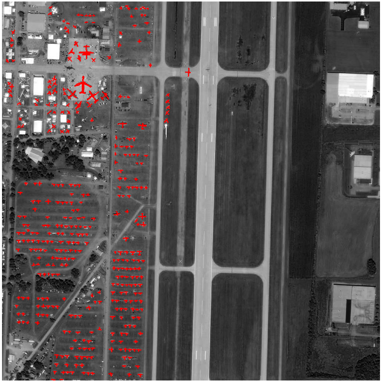
Representation - Language

- Language components offer different information
- Derivative components can offer "new information" and reduce search space
- Example: Simple (left) vs Derived (right)

- sub(x,y)
- mult(x,y)
- max(x,y)

- Edge Filters
- Geometric Filter
- ...

Representation



Fitness Function

- Produces values on which to compare generated solutions
- Defines the "desired solution"
- Fitness function will therefore impact performance

Figure: f(x) = Max(true positives)

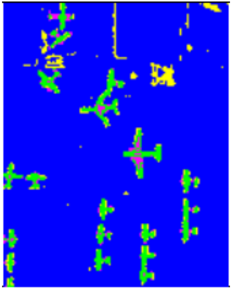


Figure: f(x) = Min(false positives + false negatives)

Applications

- NERO - Evolved Neural Networks
- Starcraft 2 build orders - Genetic Algorithm
- Evolved Antenna - Evolutionary Algorithm
- Boxcar 2D - Genetic Algorithm
- Evolving Virtual Creatures

No Free Lunch theorem

No Free Lunch - Wolpert/Macready 1997

For all possible performance measures, no search algorithm is better than another when its performance is averaged over all possible discrete functions.

- No silver bullet
- Does not state that algorithms cannot generalize to categories of problems

Optimizing Optimizers

Optimizing Optimizers

- Metaheuristics are general because they can be tuned to a specific problem
- Tuning is itself an optimization problem
- Trading one search space for another

Apply Another Metaheuristic

- Same problem exists for the higher-level heuristic
- computation increases exponentially

Questions?