Genetic Algorithms and Multi-Objective Genetic Algorithms

Luíño Seoane

June 12, 2013

Outline

Brief history

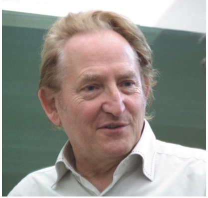


Figure: **Ingo Rechenberg**

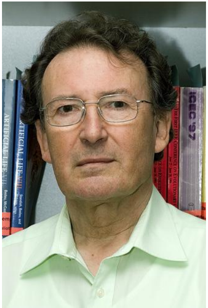
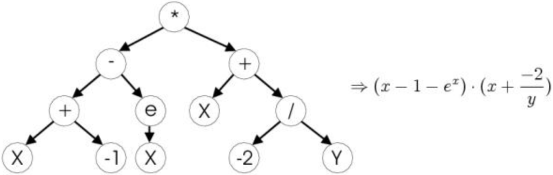


Figure: **Hans Paul Schwefel**

Figure: **Lawrence J. Fogel**



- Genetic Programming.
- Loads of applications on engineering.

Brief history

```
A  =  0 1 1 1 0 0 0
H1 =  * 1 * * * * 0
H2 =  * * * 1 0 * *
```

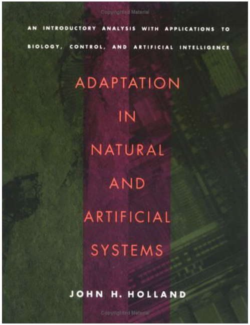
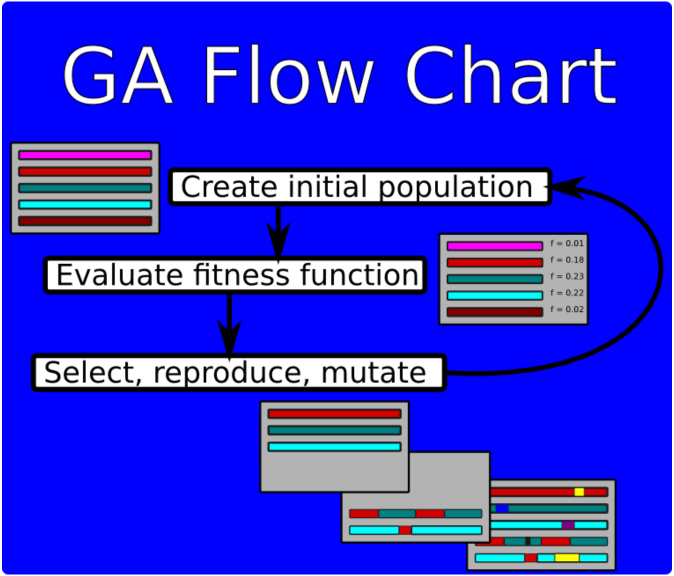


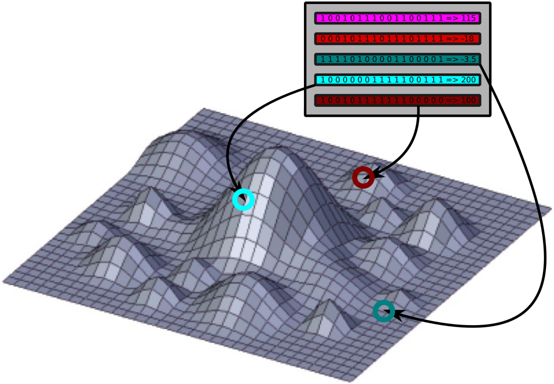
Figure: **John H. Holland**

Outline

General framework



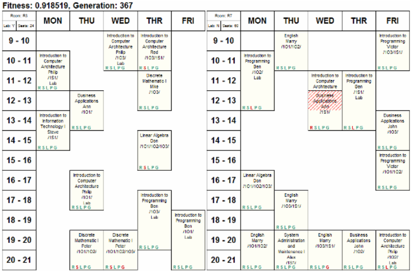
Scope:



Optimizing analytical functions

- Heuristic method.
- Encoding solutions as bit strings: easy to handle, easy to mutate.

Scope:



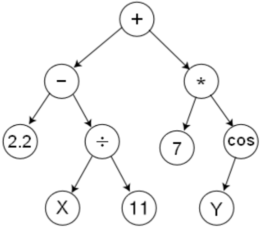
Combinatorial problems

- Usually hard (NP).
- Find proper encoding.
- Crossover and mutation must preserve structure.

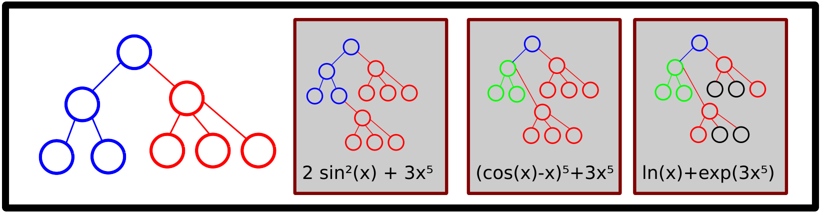
Scope:

Genetic Programming

- Functions can be described as trees.
- Also programs.
- Rube Goldberg machines.



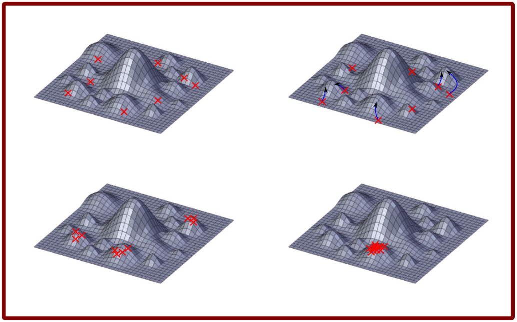
$$\left(2.2 - \left(\frac{X}{11}\right)\right) + \left(7 * \cos(Y)\right)$$



Problems:

Important problems

- Convergence to local optima.
- Loss of diversity.
- Can't cope with dynamic optimization.



Minor problems

- Complex fitness function.
- Scaling of search space.

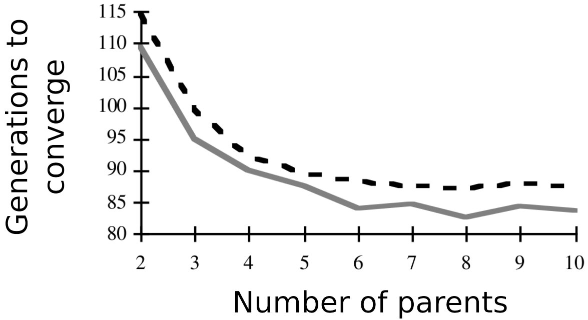
Standard strategies

Fitness function and selection

- Link between GA and problem.
- Fitness to whole chromosomes or to individual genes.
- Selection of parents:
  - Biologically inspired: $p(\mathbf{x}_i) = \frac{f(\mathbf{x}_i)}{\sum_j f(\mathbf{x}_j)}$
  - Tournament.
  - Select fittest.

Multi-parenting: orgy in the computer

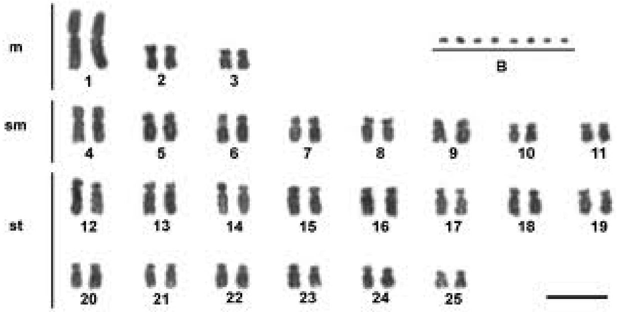
- Two scenarios:
  - Exponential decay of performance increase.
  - Few parents perform better (combinatorial).



Standard strategies

Diploids, Multiplods, Structured GA

- Encoding implicit memory.
- Dominances must be resolved:
  - External, fitness based, coded within genes...
- Neutral mutations enhance exploration.



Standard strategies

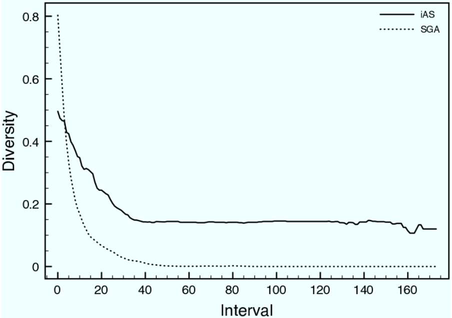
Dynamic systems: increasing diversity

- Diploids, multiploids, structured GA.
- Penalize fitness function:
  - Based on local abundance of solutions.
- Random inmigrants.
- Explicit memory of good solutions:
  - Release them by environment change.

- Usually harms convergence.

Alternative Splicing [4]

- Similar to structured GA.
- Promotes neutral mutations.
- Good reaction to environmental changes.
- Keeps diversity.

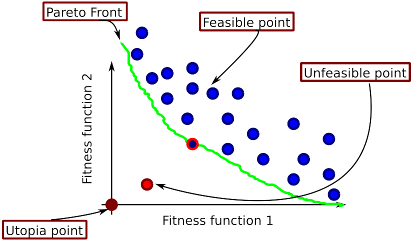


Outline

Multiple objective optimization

What is the solution??

- We don't seek one only individual.
- Instead: a method to control trade-offs between objective functions.
  - Pareto front of non-dominated solutions.
  - Solutions should sample the whole space of optimals.



- $x \succ y \Leftrightarrow$
  - $f_k(x) \leq f_k(y), k = 1, ..., K$
  - $\exists k' \in \{1, ..., K\}, \;\; f_{k'}(x) < f_{k'}(y)$.

Multiple objective optimization

How to cope with this?

- Optimize by parts?
  - Subpopulations for different objectives that are mixed.
  - Much stronger local optima than usual.
- Combine fitness functions:
  - Are they commensurable?
  - What is the proper combination:
    - $F(\mathbf{x_i}) = \sum_j \alpha_j f_j(\mathbf{x}_i)$?
    - $F(\mathbf{x_i}) = \prod_j [f_j(\mathbf{x}_i)]^{\alpha_j}$?
  - Introduces *ad-hoc* constrains.

How to cope with this?

- Try weighted fitness functions once more:
  - Encode the weights within the genome.
  - Evaluate the solutions with random weights.
  - Drawbacks:
    - Doesn't explore the trade-off surface.
    - Crashes if Pareto front is not convex.

Pareto-ranking-based optimization

- Assign fitness depending on dominance level.
  - Different implementations.
  - Natural ways of penalizing close solutions.