

Genetic Algorithms and Multi-Objective Genetic Algorithms

Luíño Seoane

June 12, 2013

- 1 Some toy examples and very beginnings
- 2 Genetic Algorithms – General strategies
- 3 Multi-Objective Genetic Algorithm

- 1** Some toy examples and very beginnings
- 2 Genetic Algorithms – General strategies
- 3 Multi-Objective Genetic Algorithm



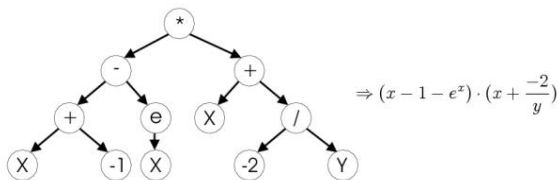
Figure: Ingo Rechenberg



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 * *

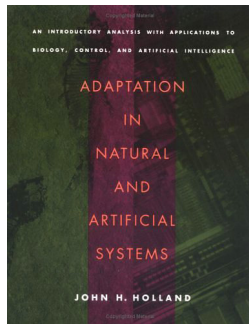
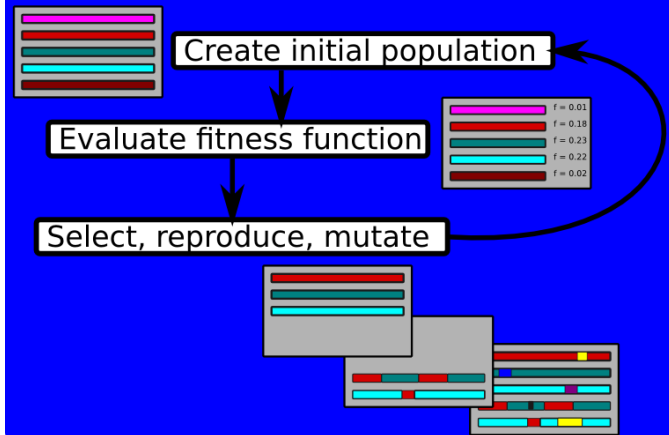
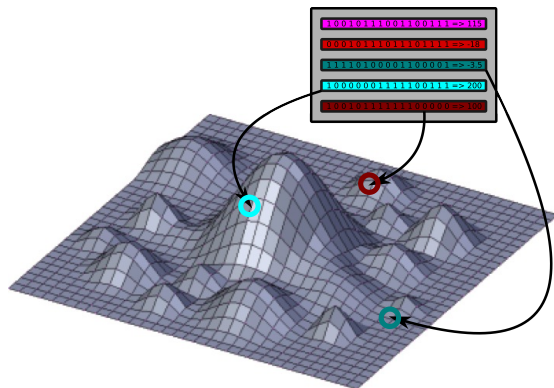


Figure: John H. Holland

- 1 Some toy examples and very beginnings
- 2 Genetic Algorithms – General strategies
- 3 Multi-Objective Genetic Algorithm

GA Flow Chart





Optimizing analytical functions

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

Fitness: 0.918519, Generation: 367

Room No	MON	THU	WED	THR	FRI	Room No	MON	THU	WED	THR	FRI
Lab 1	Lab 2	Lab 3	Lab 4	Lab 5	Lab 6	Lab 7	Lab 8	Lab 9	Lab 10	Lab 11	Lab 12
9 - 10			Introduction to Computer Architecture (P10) / I10 / I10	Introduction to Computer Architecture (P10) / I10 / I10		9 - 10		English (P10) / I10 / I10			Introduction to Programming (P10) / I10 / I10
10 - 11	Introduction to Computer Architecture (P10) / I10 / I10					10 - 11	Introduction to Programming (P10) / I10 / I10				
11 - 12				Discrete (Mathematic) (P10) / I10 / I10		11 - 12			Introduction to Computer Architecture (P10) / I10 / I10	Introduction to Programming (P10) / I10 / I10	
12 - 13		Business Applications (P10) / I10 / I10				12 - 13			Business Applications (P10) / I10 / I10		
13 - 14	Introduction to Information Technology (P10) / I10 / I10					13 - 14					Business Applications (P10) / I10 / I10
14 - 15				Linear Algebra (P10) / I10 / I10		14 - 15					
15 - 16						15 - 16					Introduction to Programming (P10) / I10 / I10
16 - 17		Introduction to Computer Architecture (P10) / I10 / I10				16 - 17	Linear Algebra (P10) / I10 / I10				
17 - 18				Introduction to Programming (P10) / I10 / I10		17 - 18		English (P10) / I10 / I10			
18 - 19					Introduction to Programming (P10) / I10 / I10	18 - 19					
19 - 20	Discrete (Mathematic) (P10) / I10 / I10	Discrete (Mathematic) (P10) / I10 / I10				19 - 20	English (P10) / I10 / I10	System Administration and Networking (P10) / I10 / I10	English (P10) / I10 / I10	Business Applications (P10) / I10 / I10	Introduction to Computer Architecture (P10) / I10 / I10
20 - 21						20 - 21					

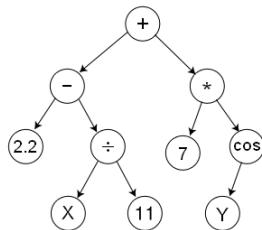
Combinatorial problems

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



Genetic Programming

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



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

2 $\sin^2(x) + 3x^5$

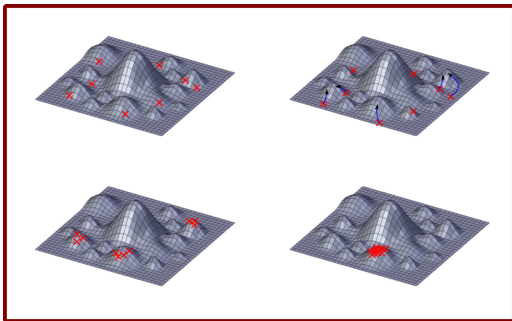
$(\cos(x)-x)^5+3x^5$

$\ln(x)+\exp(3x^5)$

Problems:

Important problems

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



Minor problems

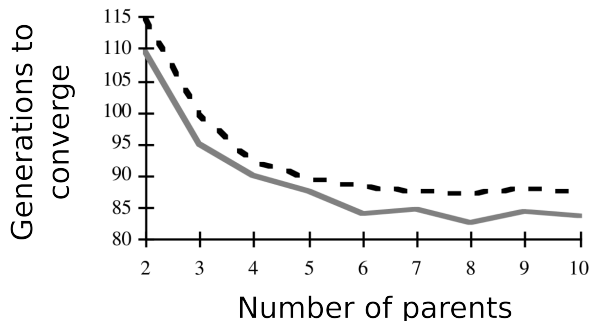
- Complex fitness function.
- Scaling of search space.

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.

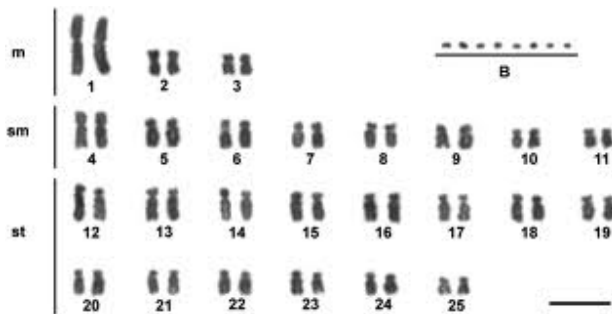
Multi-parenting: orgy in the computer

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



Diploids, Multiplods, Structured GA

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

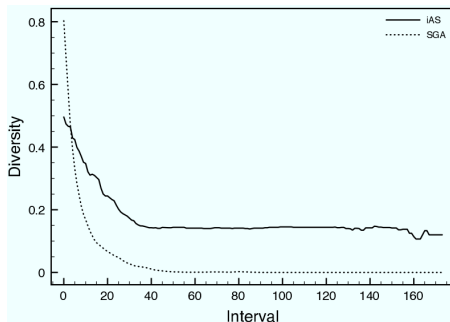


Dynamic systems: increasing diversity

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

Alternative Splicing [4]

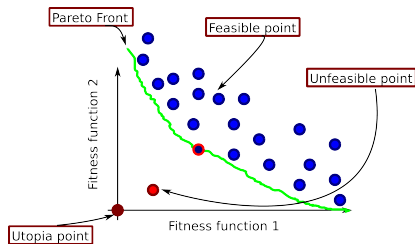
- Similar to structured GA.
- Promotes neutral mutations.
- Good reaction to environmental changes.
- Keeps diversity.



- 1 Some toy examples and very beginnings
- 2 Genetic Algorithms – General strategies
- 3 Multi-Objective Genetic Algorithm**

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.



$$\bullet x \succ y \Leftrightarrow$$

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

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.

