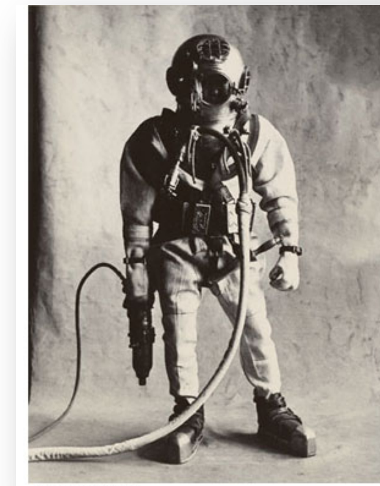
Advanced Evolutionary Algorithms: An Introduction

Luis Martí Orosa

LIRA/DEE/PUC-Rio

Why we (must) study AI?



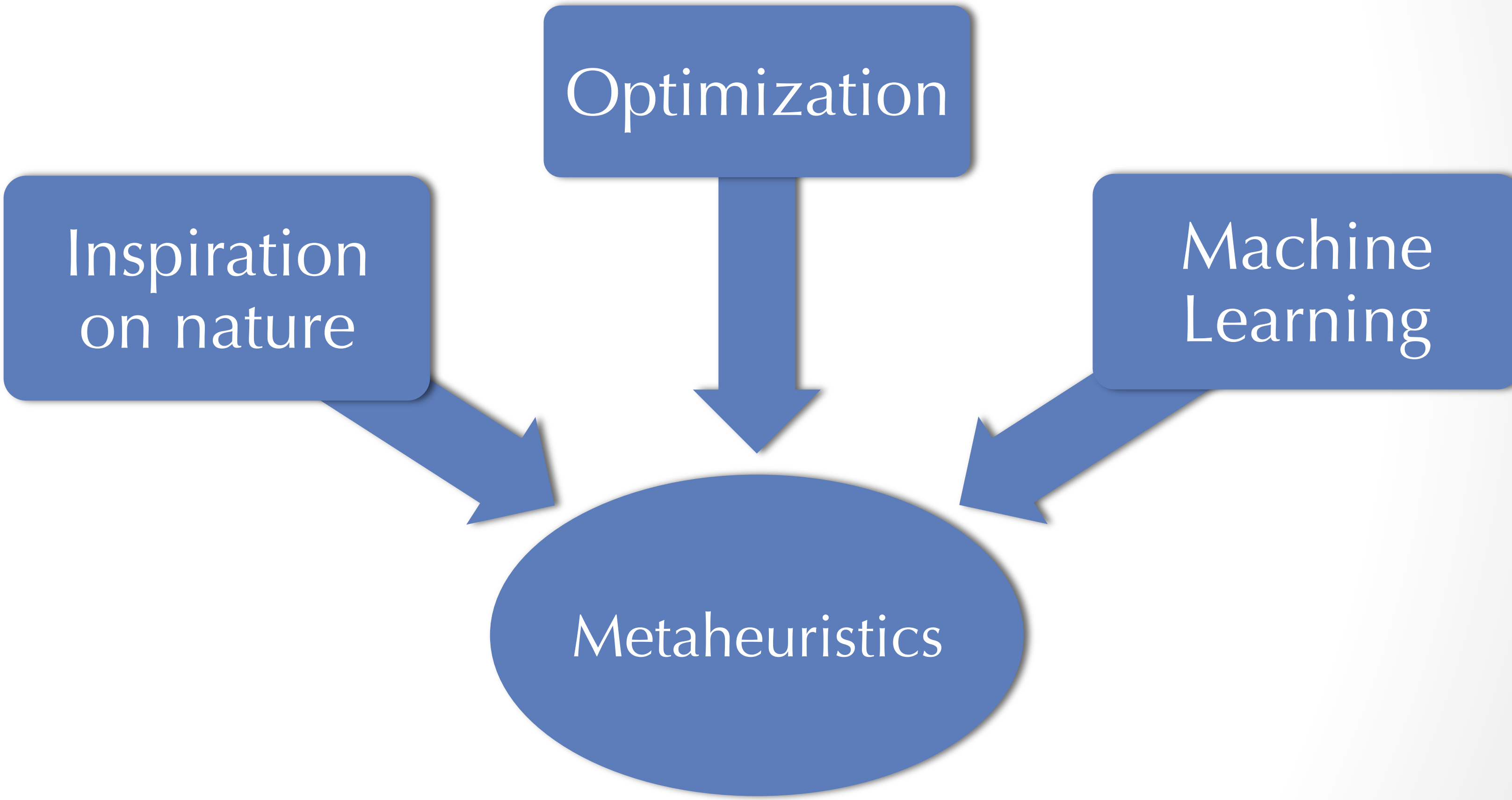
Intelligence defines us…

…and it would be “great” to have machines that share that feature.

Inspired by nature

- Neural networks.
- Genetic and evolutionary algorithms.
- Swarm-based approaches.
- Artificial immune systems
- Ants/bees (and other bugs) colonies and emergent behaviors.
- Etc…

Nature-inspired metaheuristics



Origin

- Throughout history we have been fascinated with life.
- The origin the explanation of life is present in ancient texts.
- In "western" countries we have an Abrahamic tradition.
  - Life has been **created** at a given time with a **purpose**.
  - The purpose of life has to do with the human beings.
  - Everything in nature (and therefore life) is ruled by a "supreme entity".

… but fossils were discovered.

Converging schools of thought

Jean-Baptiste Lamarck

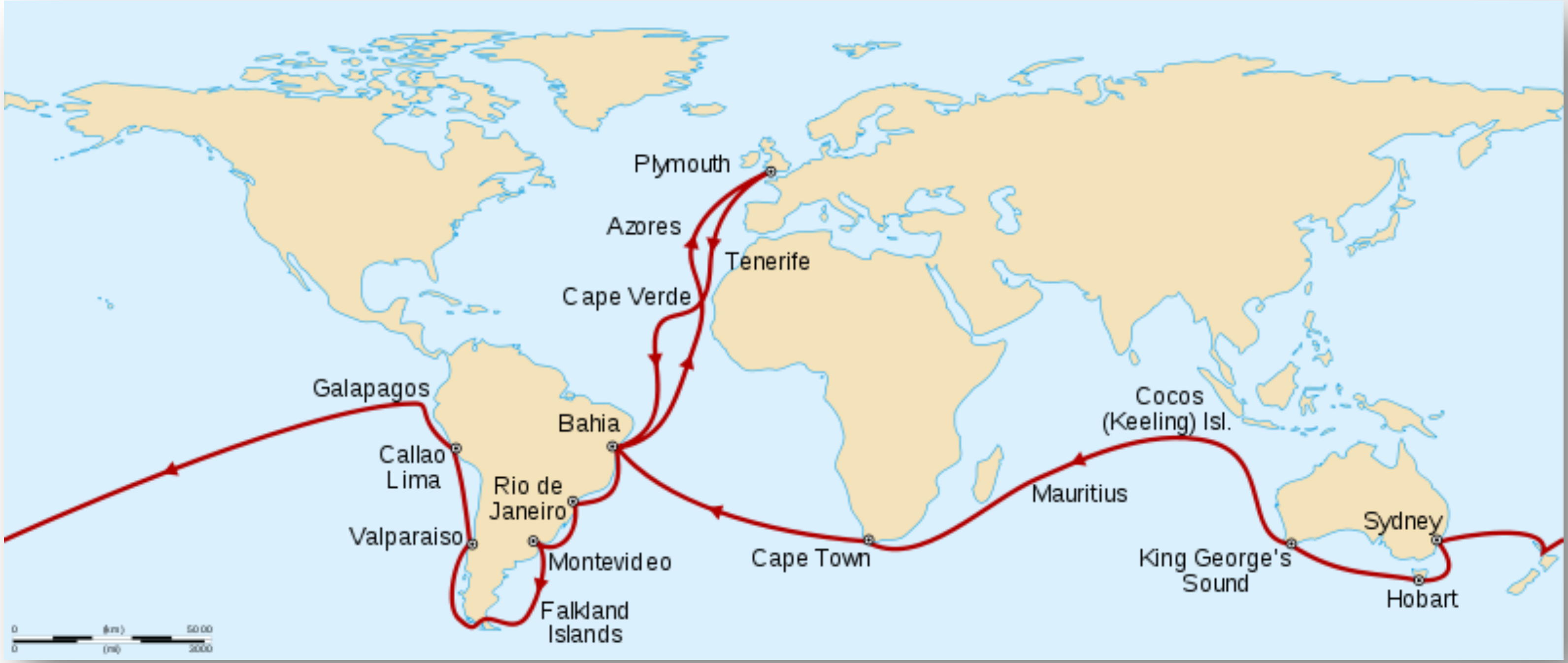
- Species descend one from other.
- Characteristics acquired during life are passed to offspring.

Erasmus Darwin

- Organic evolution (similar to Lamarck).
- Introduces the notion of competition for resources and matting.

Rev. Thomas Robert Malthus.

Charles Darwin in the Beagle



- During the trip he focused in geology.
  - *Structure and distribution of coral reefs* (1842)

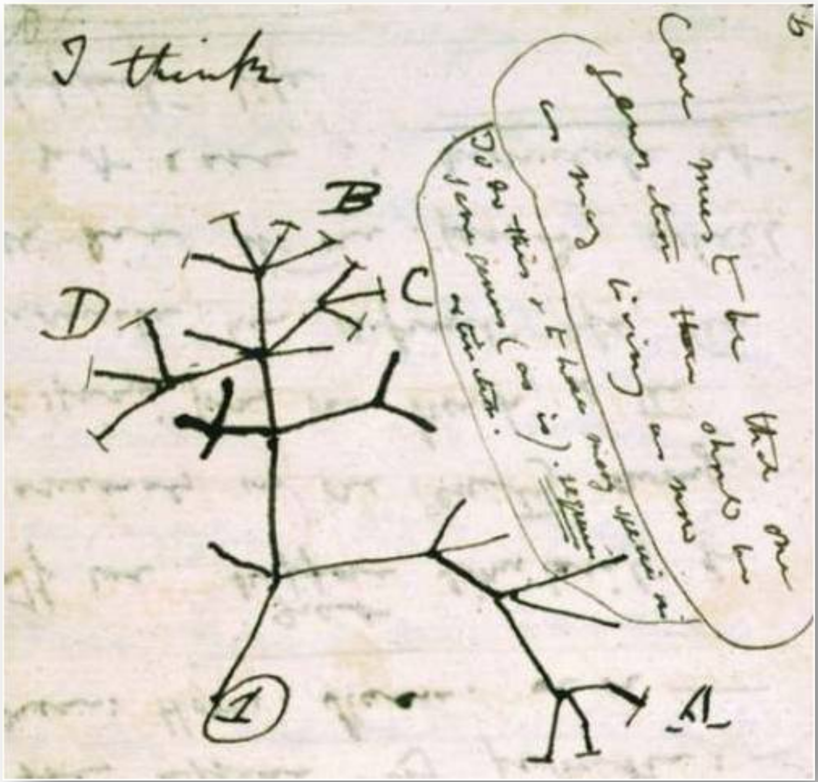
Conclusions not from the trip

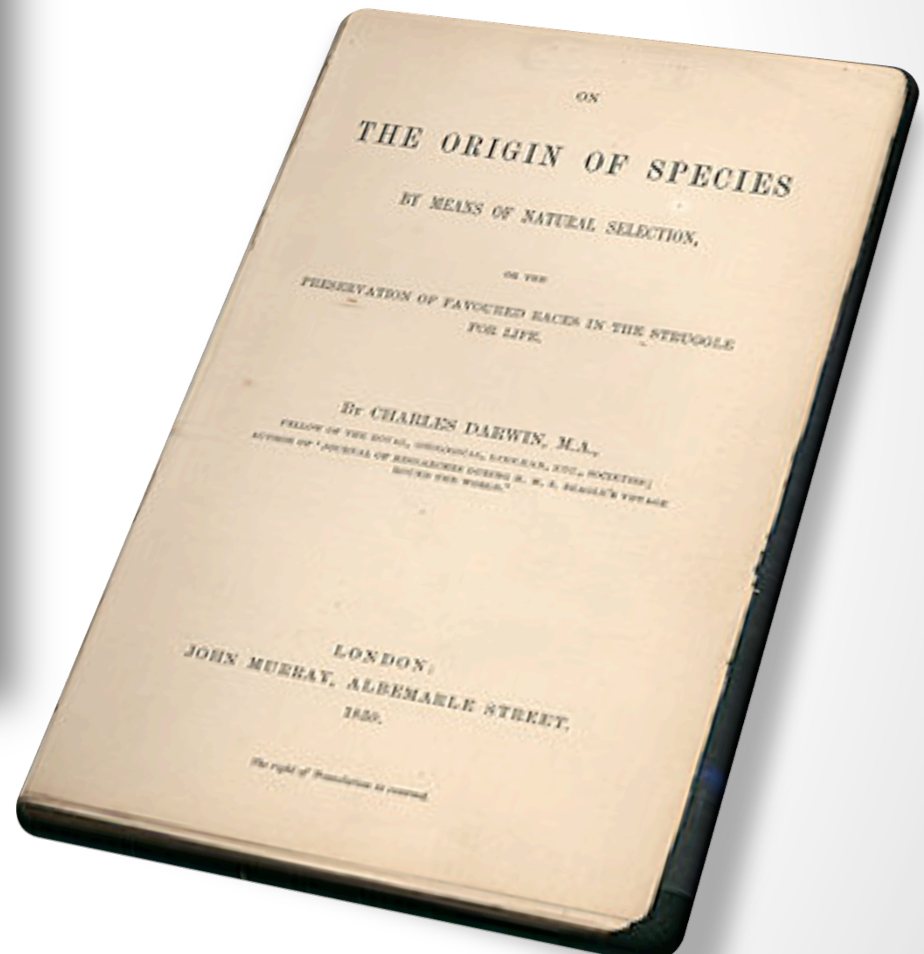
Ornithology

Alfred Rusell Wallace

- While Darwin was busy “in his garden”, **Wallace** (1823-1913) reached to the same conclusions
  - He was in the Malay Islands.
  - Derived the theory of evolution from Malthaus self-regulation of human population studies.
- In 1858 Darwin and Wallace published together their theory.

The Origin of Species […]



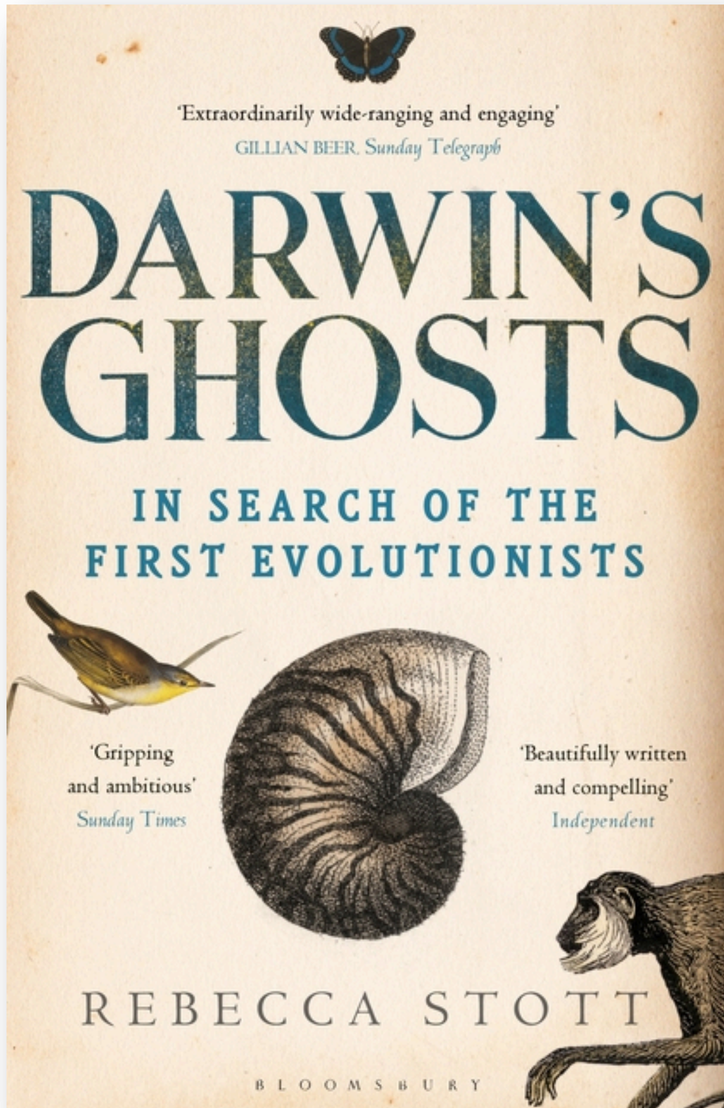


Theories derived

1. Evolution.
2. Common ancestors.
3. Multiplication of species.
4. Gradualism.
5. Natural selection.

Darwin was a trending topic!

Darwin was not the only one



Optimization

Optimization

A key subject:

- Computer Science
- Artificial Intelligence
- Operations Research
- Engineering
- …

To optimize is an imprecise term that essentially means "make better"

Optimization

- “The process of finding the best solution for a given optimization problem with a given resource and temporal budget.”
- Optimization problem:
  - Has a number of feaseble solutions.
  - There is a clear notion of quality of solutions.
  - The best solution: global optimum

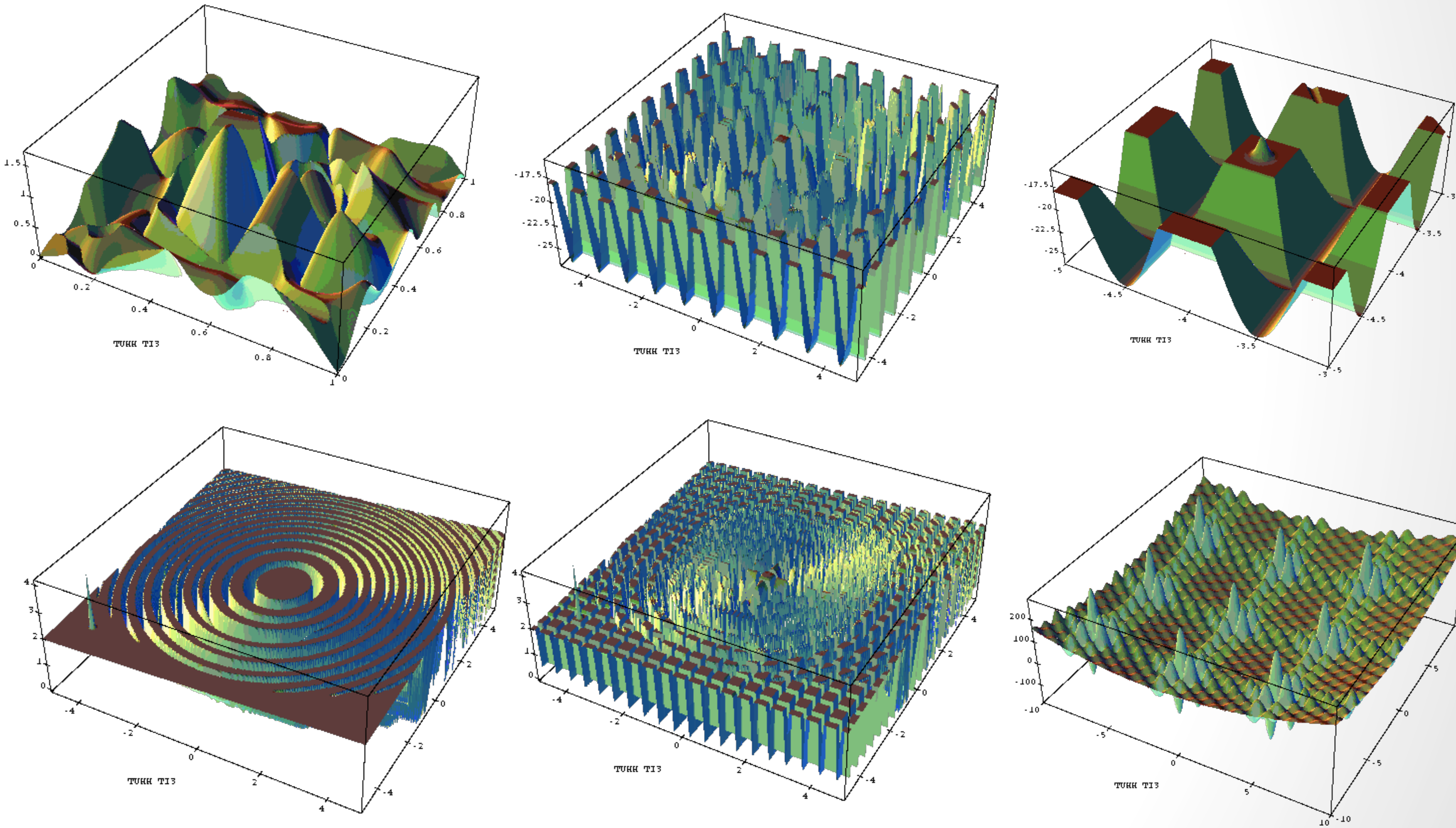
“Hard” and “easy” problems

- **Tratable**: if there is an algorithm that solves it polinomial time.
- **Intratable (hard)**: if there is no algorithm that solves the problem in polinomial time, NP problems.

Problemas de Optimización

- We are interested in “hard problems”
- Not warrantied that the solution can be found.
- Properties of the problem is unknown.
- We need metaheuristics:
  - “Reduced” computational complexity.
  - Do not ensure convergence to the global optimum.

Optimization problems



Classes of problems and solvers

- Optimization problems:
  - Combinatorial vs numerical
  - Single vs multi-objective
- Algorithms:
  - Exact:
    - Linear programming
    - Dynamic programming
    - "Branch-and-bound"
  - Approximated or (meta)heuristic

Iterative Stochastic Methods

1. Start:
   - Generate and evaluate an initial collection of candidate solutions, S.
2. Production:
   - Select elements of S. Produce and evaluate a new set of candidate solutions S′ by means of modifications of the selected elements.
3. Replacement:
   - Replace some elements of S with some elements of S′ and return to 2.

Why are these methods used?

- Easy to explain and implement.
  - A few lines of pseudocode describe the essential elements of most of these algorithms.
- They are multi-purpose.
  - Do not have strong a priori requirements.
- Proven success.
- Easy to adapt to particular problems with problem dependent (local) methods.

Montecarlo search

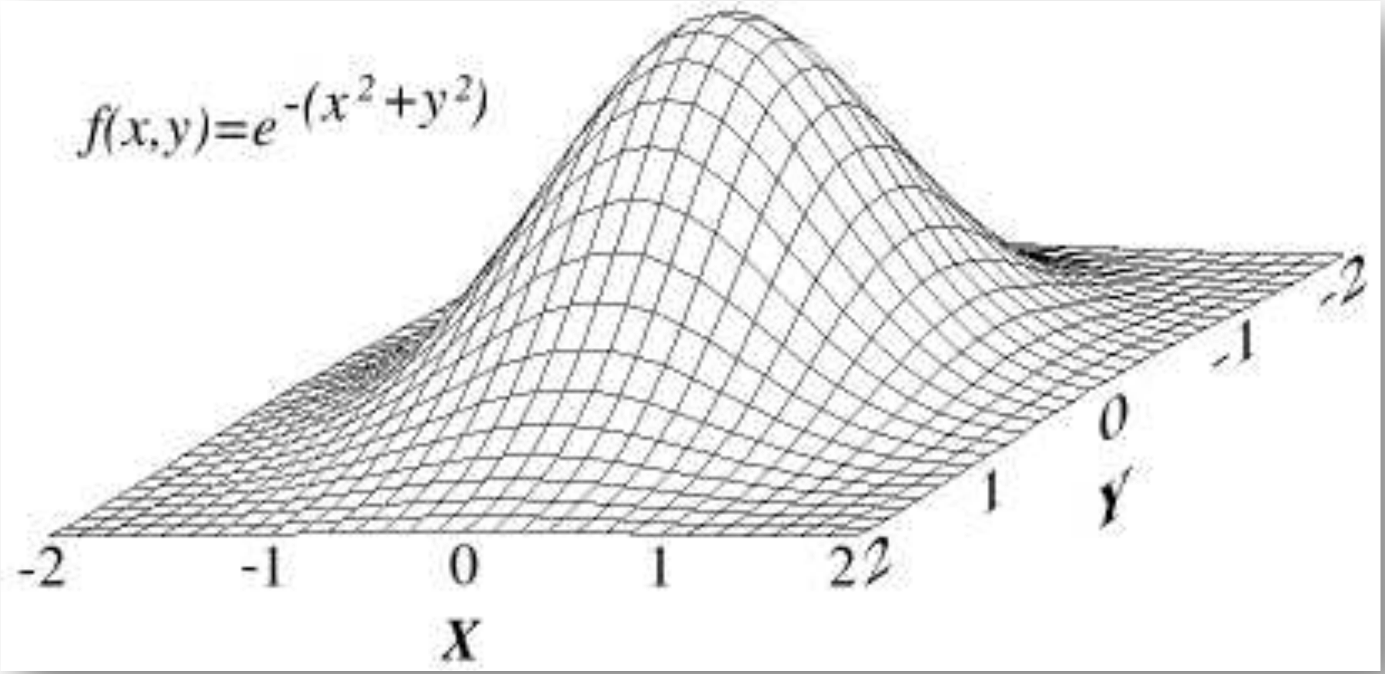
```
t = 0;
result = createNewSolution();
evaluate(result);
while notFinished() do
  a = createNewSolution();
  evaluate(a);
  if a isBetterThan result then
    result = a;
 t = t+1;
end_while
```

Hill-Climber

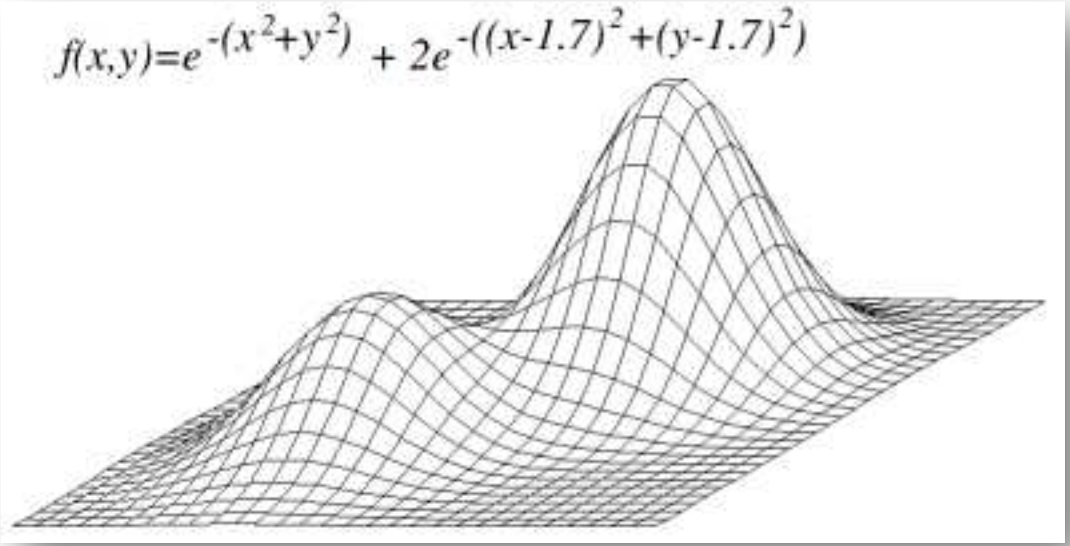
```
t = 0;
result = createNewSolution();
evaluate(result);
while notFinished() do
  a = clone(result);
  mutate(a);
  evaluate(a);
  if a isBetterThan result then
    result = a;
 t = t+1;
end_while
```

Sample problem Montecarlo/H.C.

- Only one (global) optimum.



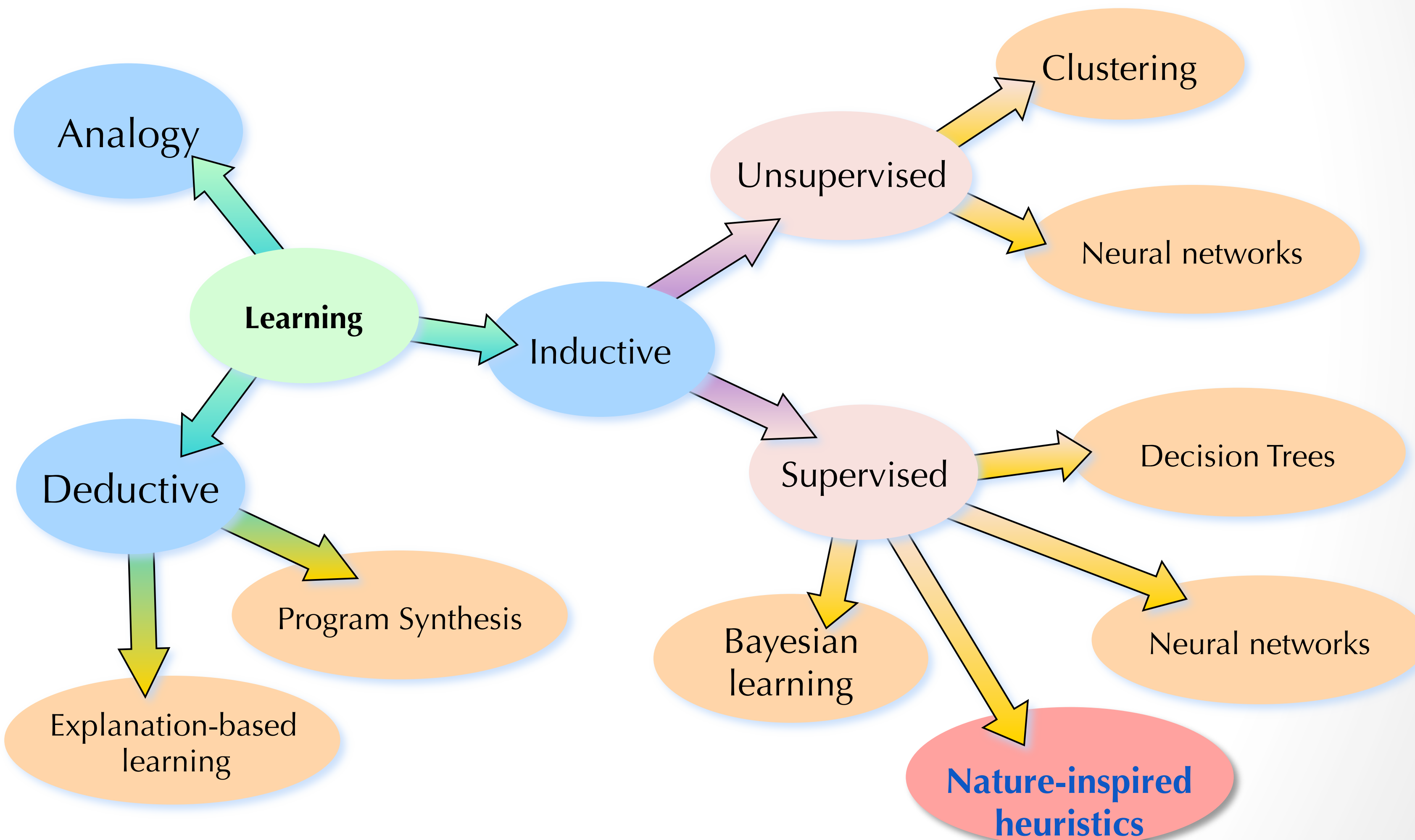
Local optima



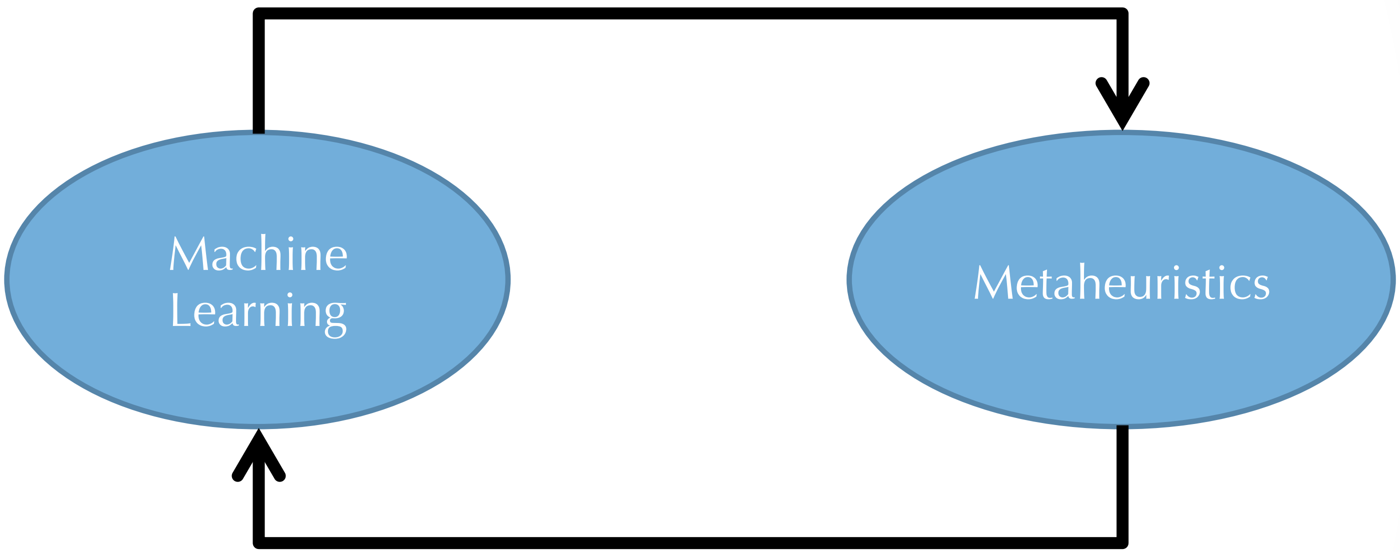
Repeat the algorithm with different initializations.

Nature-inspired methods

AI/Machine Learning

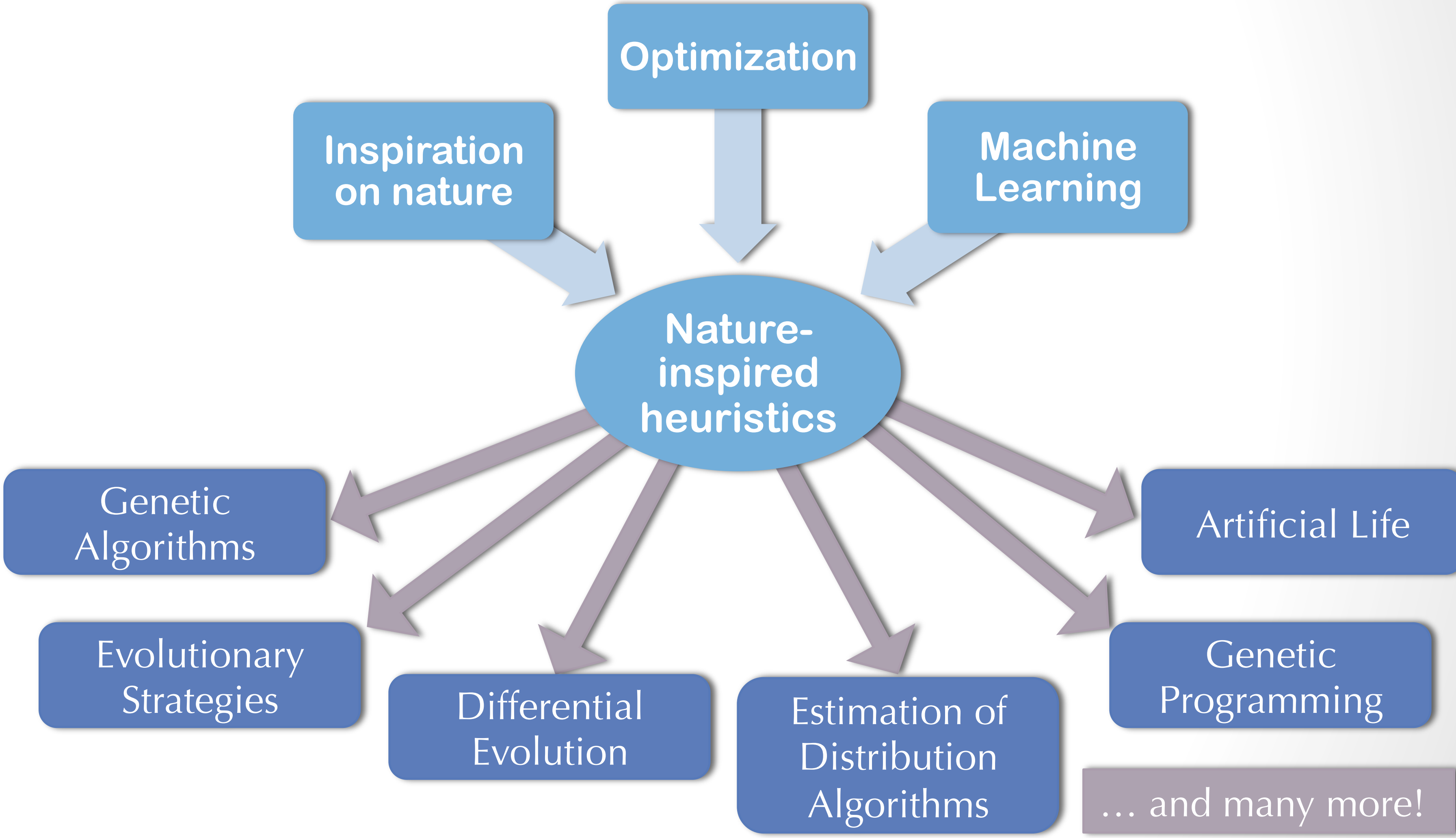


Metaheuristics and ML



Machine learning approaches exploit metaheuristics and viceversa.

Nature-inspired metaheuristics



Nature's optimization algorithm?

- Feasible solutions are represented as a string (ADN).
- Populations of solutions.
- Evaluates every solution (individual) and eliminates the worst.
- Natural selection thanks to the survival of the fittest.
- New population combines surviving individuals:
  - Crossover.
  - Mutation.
- Repetition and lots of time.

Evolutionary computing

- Computational simulation of the processes of evolution and natural selection.
- Mainly inspired by the theory of evolution.
  - Require little information of the problem
  - General purpose
  - Can contain/cooperate with other methodologies
  - Can be used in an "interactive mode".
  - Inherent parallelism.
  - Robust with respect to data.

Aplications of EC

- As an engineering tool, for finding solutions in optimization problems.
  - Combinatorial and numerical optimization.
  - Planning and control
  - Engineering design
  - Data mining and machine learning applications.
- As a science tool
  - Simulation of real-world phenomena: artificial life, cellular automata, directed evolution, etc.

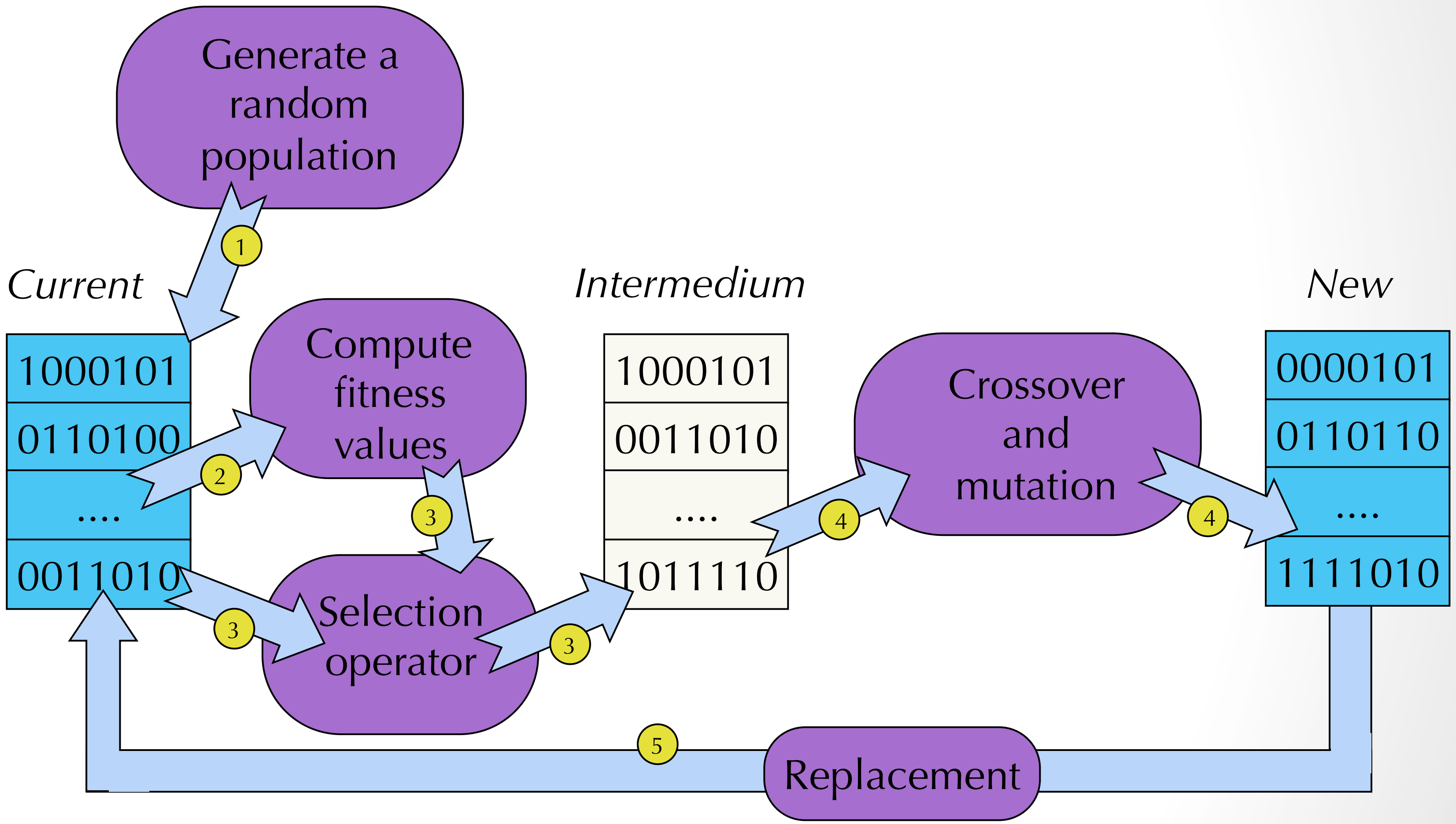
Search

- Search space: set of all possible solutions.
  - Its size is an indicator of problem complexity.
- Crossover operator: combines characteristics of two or more individuals – local search.
- Mutation: generates new individuals with different characteristics – global search.
- Together they implement a pseudo-random walk:
  - Random, as operators are not deterministic.
  - Directed; as selection is controlled by the fitness function that tends to improve the quality of solutions.

Genetic Algorithms

- **John Holland**, 1960s
- “Adaptation in natural and artificial systems”, 1975.

Genetic Algorithm process



Using GAs

- Modeling the problem
  1. Decide how to encode information.
  2. Create fitness function.

- This is a key part! -

- Configure GAs
  1. Matting selection and replacement selection.
  2. Type of crossover and mutation.
  3. Parameters

Selection Schemes

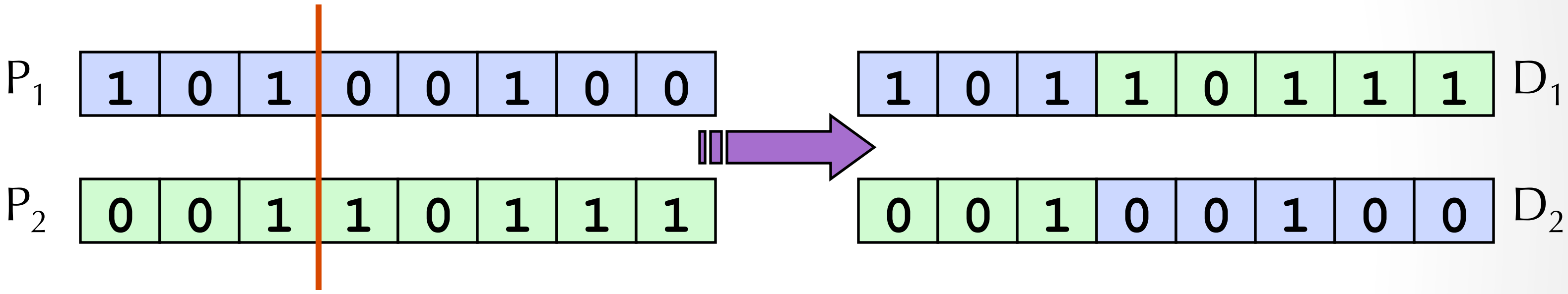
Roulette wheel selection

- Proportional to fitness or ranking.

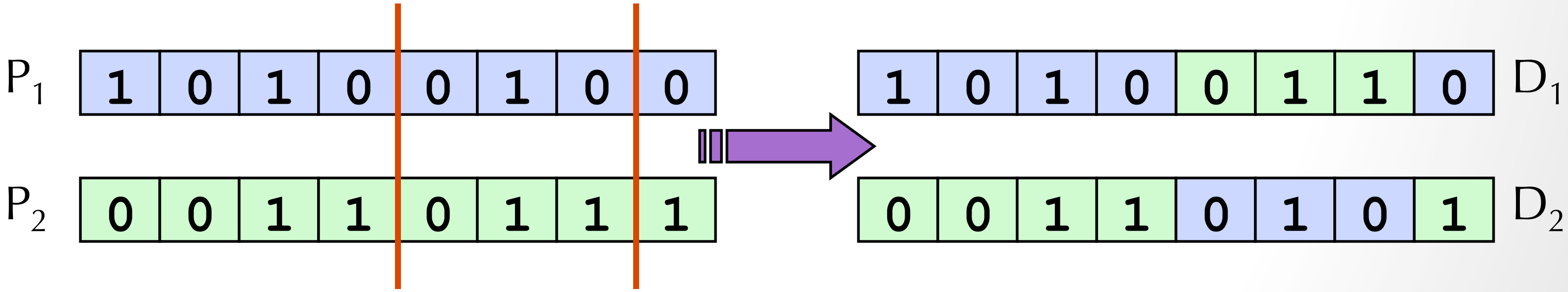
- stochastic sampling
    - roulette wheel selection
    - spin wheel N times
- stochastic universal sampling
    - roulette wheel selection
    - single spin, wheel has N equally spaced markers
- tournament selection
    - choose *k* candidates at random with uniform probability
    - pick best one for reproduction

Crossover

- 1 point cross-over



- 2-points cross-over

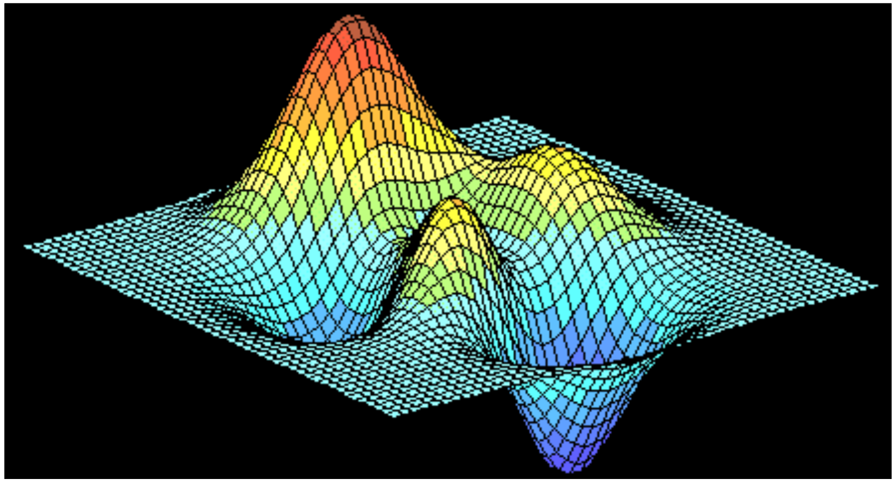


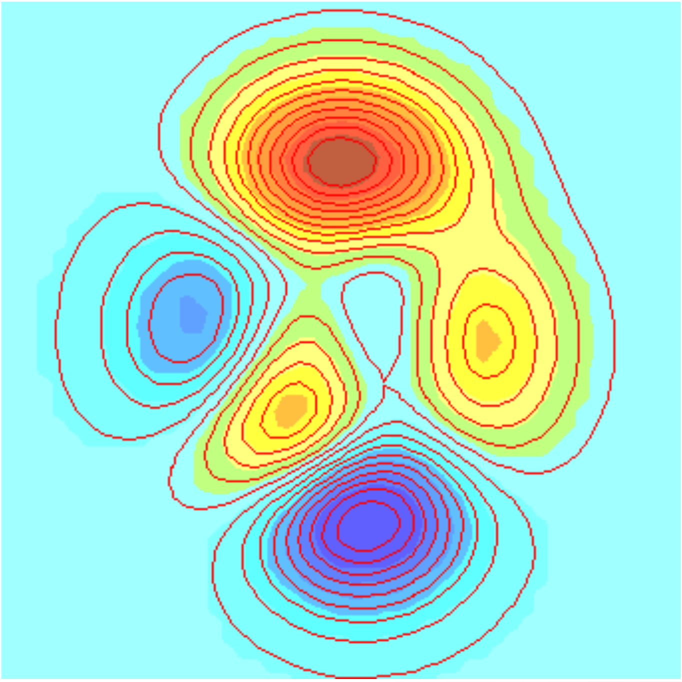
Mutation

- Every gene is examined.
- An allele is mutated with a low probability, p_m (0.001-0.1)%

GAs at work

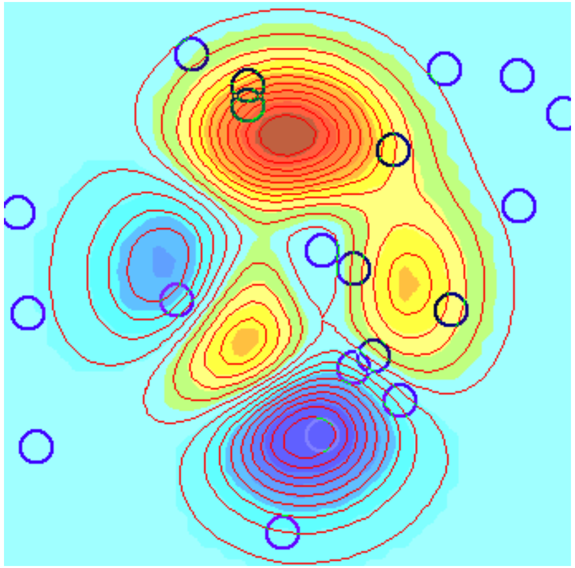
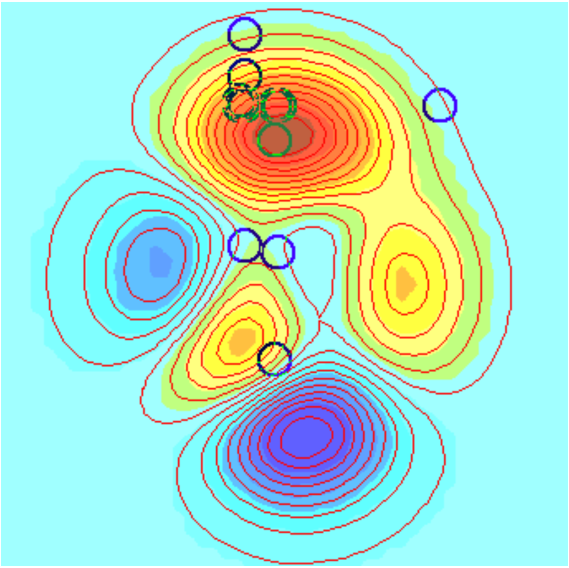
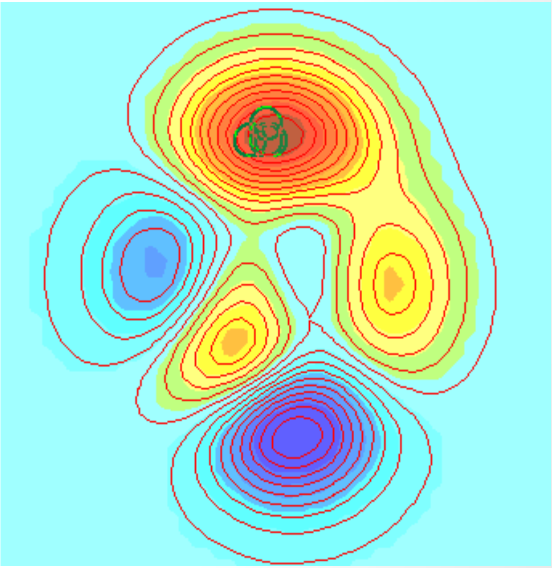
$$z = f(x,y) = 3(1-x)^2 e^{-\left(x^2+(y+1)^2\right)} - 10\left(\frac{x}{5} - x^3 - y^5\right)e^{-\left(x^2+y^2\right)} - \frac{1}{3}e^{-\left((x+1)^2+y^2\right)}$$





GAs at work

- Population as iterations advance
  - $t = 0$
  - $t = 5$
  - $t=10$

Advanced GAs

- Diploid crossover.
- Multi-objective approaches.
- Knowledge-based methods.
- Multiple populations
- Coevolution
- Parallelization

http://boxcar2d.com/

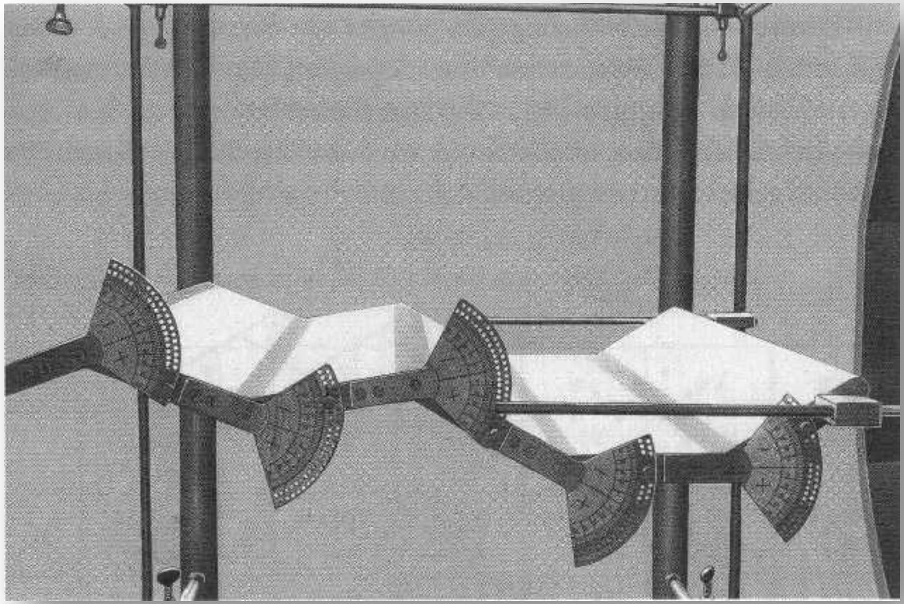
Evolutionary Strategies

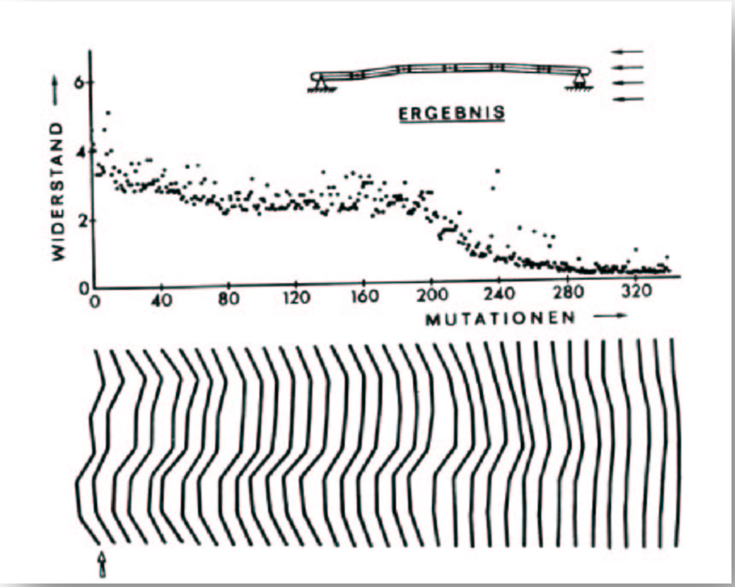
Evolutionary Strategies

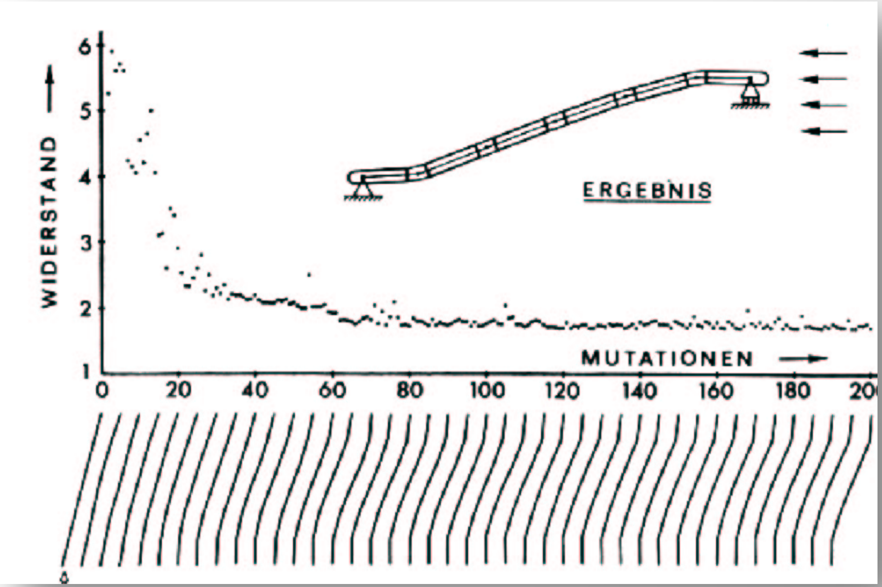
- Proposed in the 60s by Rechenberg and Schwefel.
- Method of parametric (numeric) optimization.
- Only mutation, with self-adaptation.
- Classes:
  - Simple EE (population of two)
  - Multiple EE (more elements)
- Characteristics
  - Fast
  - Solid theoretical foundation
  - Good results in numerical optimization.

Initial steps

- Airfoil profile







ES technical summary tableau

Representation	Real-valued vectors
Recombination	Discrete or intermediary
Mutation	Gaussian perturbation
Parent selection	Uniform random
Survivor selection	(μ,λ) or $(\mu+\lambda)$
Specialty	Self-adaptation of mutation step sizes

Simple (1,1) Pseudocode

```
Set t = 0
Create initial point x^t = ⟨ x_1^t,…,x_n^t ⟩
repeat
  Draw z_i from a normal distr. for all i =
  1,…,n
  y_i^t = x_i^t + z_i
  IF f(x^t) < f(y^t) THEN
    x^(t+1) = x^t
  ELSE
    x^(t+1) = y^t
  END_IF
  Set t = t+1
until endCondition()
```

Representation

- Chromosomes consist of three parts:
  - Object variables: $x_1,\ldots,x_n$
  - Strategy parameters:
    - Mutation step sizes: $\sigma_1,\ldots,\sigma_{n_\sigma}$
    - Rotation angles: $\alpha_1,\ldots, \alpha_{n_\alpha}$
- Not every component is always present
- Full size: $\langle x_1,\ldots,x_n, \sigma_1,\ldots,\sigma_n ,\alpha_1,\ldots, \alpha_k \rangle$
- where $k = n(n-1)/2$ (no. of i,j pairs)

Mutation

- Main mechanism: changing value by adding random noise drawn from normal distribution
- $x'_i = x_i + N(0,\sigma)$
  - σ is part of the chromosome $\langle x_1,\ldots,x_n, \sigma \rangle$
  - σ is also mutated into σ'
- Thus: mutation step size σ is coevolving with the solution x

Mutate σ first

- Net mutation effect: $\langle x, \sigma \rangle \rightarrow \langle x', \sigma' \rangle$
- Order is important:
  - first $\sigma \rightarrow \sigma'$ (see later how)
  - then $x \rightarrow x' = x + N(0,\sigma')$
- Rationale: new $\langle x', \sigma' \rangle$ is evaluated twice
  - Primary: x' is good if $f(x')$ is good
  - Secondary: σ' is good if the x' it created is good
- Reversing order would not work

Mutation case 1: Uncorrelated mutation with one σ

- Chromosomes: $\langle x_1,\ldots,x_n, \sigma \rangle$
- $\sigma' = \sigma \cdot \exp(\tau \cdot N(0,1))$
- $x'_i = x_i + \sigma' \cdot N(0,1)$
- Typically the "learning rate" $\tau \propto 1/ n^{½}$
- And we have a boundary rule $\sigma' < \varepsilon_0 \Rightarrow \sigma' = \varepsilon_0$

Mutation case 2: Uncorrelated mutation with n σ's

- Chromosomes: $\langle x_1,\ldots,x_n, \sigma_1,\ldots, \sigma_n \rangle$
- $\sigma'_i = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i\,(0,1))$
- $x'_i = x_i + \sigma'_i \cdot N_i\,(0,1)$
- Two learning rate parmeters:
  - τ' overall learning rate
  - τ coordinate wise learning rate
- $\tau \propto 1/(2\,n)^{½}$ and $\tau \propto 1/(2\,n^{½})^{½}$
- And $\sigma_i' < \varepsilon_0 \Rightarrow \sigma_i' = \varepsilon_0$

Mutation case 3: Correlated mutations

- Chromosomes: $\langle x_1,\ldots,x_n, \sigma_1,\ldots, \sigma_n ,\alpha_1,\ldots, \alpha_k \rangle$
- where $k = n \cdot (n-1)/2$
- and the covariance matrix C is defined as:
  - $c_{ii} = \sigma_i^2$
  - $c_{ij} = 0$ if i and j are not correlated
  - $c_{ij} = ½ \cdot (\sigma_i^2 - \sigma_j^2) \cdot \tan(2 \alpha_{ij})$ if i and j are correlated
- Note the numbering / indices of the α's

Correlated mutations cont'd

The mutation mechanism is then:

- $\sigma'_i = \sigma_i \bullet \exp(\tau' \bullet N(0,1) + \tau \bullet N_i\,(0,1))$
- $\alpha'_j = \alpha_j + \beta \bullet N\,(0,1)$
- $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{C}')$
  - $\boldsymbol{x}$ stands for the vector $\langle x_1,\ldots,x_n \rangle$
  - $\boldsymbol{C}'$ is the covariance matrix $\boldsymbol{C}$ after mutation of the α values
- $\tau \propto 1/(2\,n)^{½}$ and $\tau \propto 1/(2\,n^{½})^{½}$ and $\beta \approx 5°$
- $\sigma_i' < \varepsilon_0 \Rightarrow \sigma_i' = \varepsilon_0$ and
- $|\,\alpha'_j\,| > \pi \Rightarrow \alpha'_j = \alpha'_j - 2\,\pi\,\mathrm{sign}(\alpha'_j)$

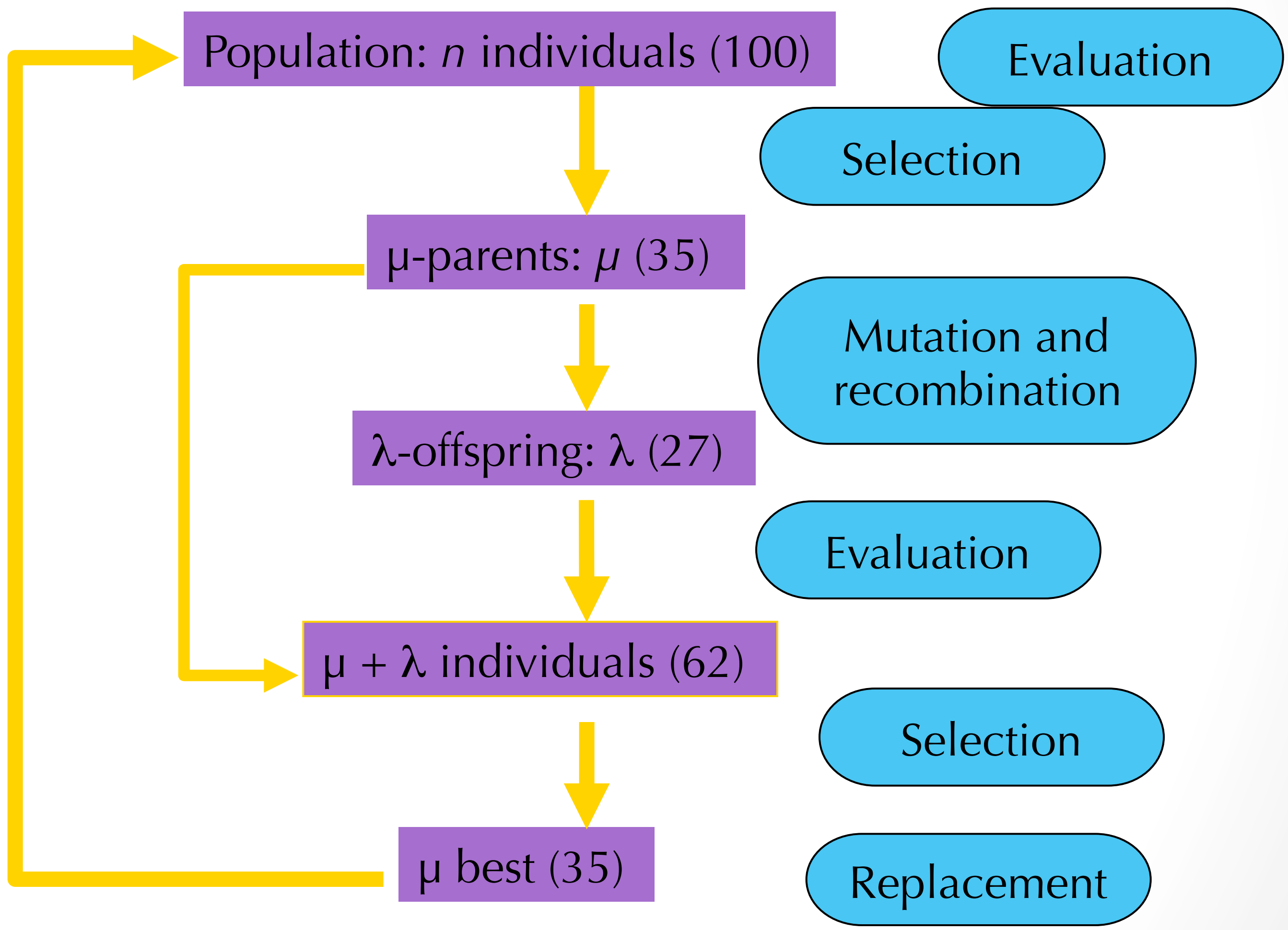
Recombination

- Creates one child
- Acts per variable / position by either
  - Averaging parental values, or
  - Selecting one of the parental values
- From two or more parents by either:
  - Using two selected parents to make a child.
  - Selecting two parents for each position.

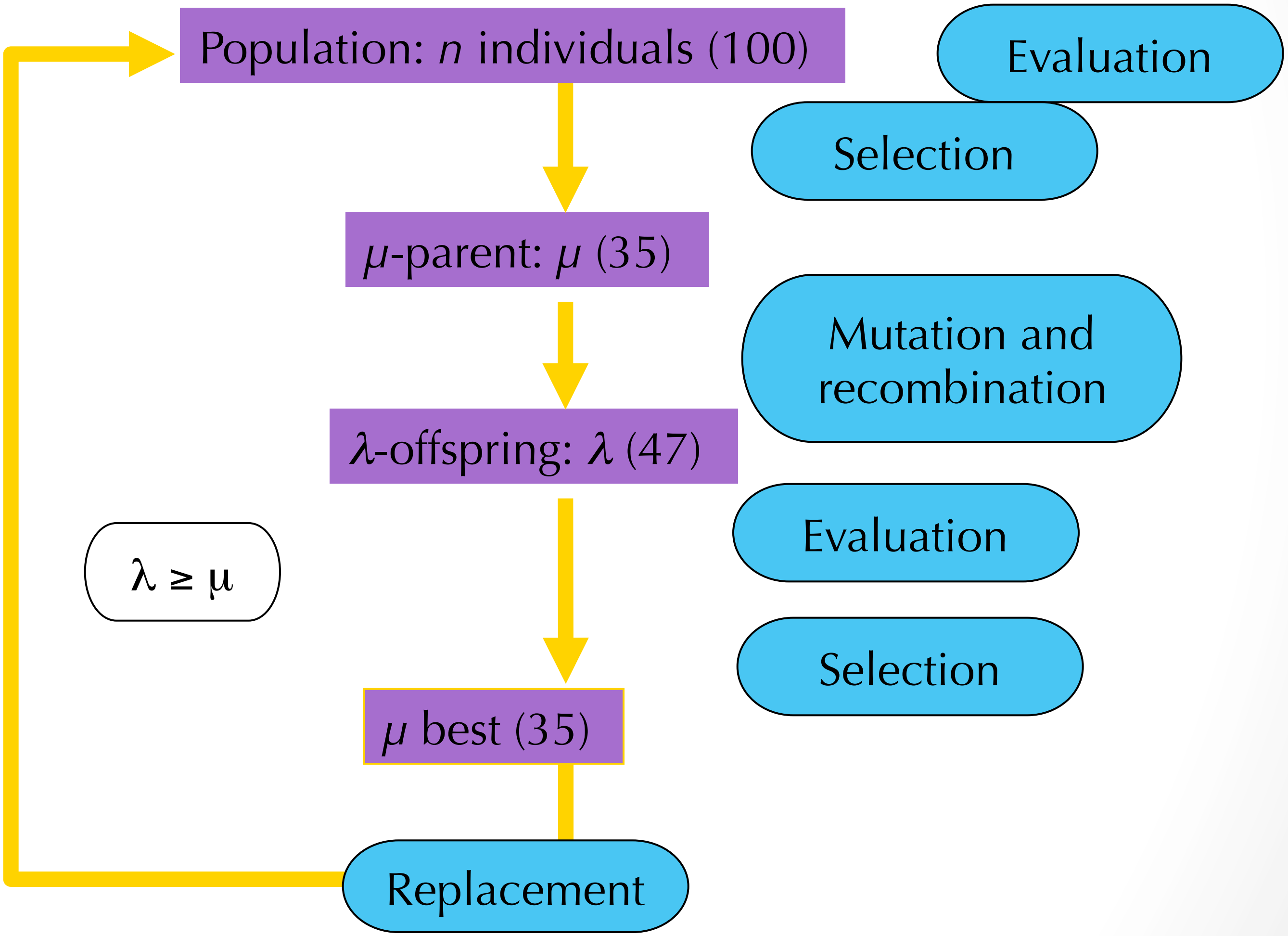
Names of recombinations

	Two fixed parents	Two parents selected for each i
$z_i = (x_i + y_i)/2$	Local intermediary	Global intermediary
z_i is x_i or y_i chosen randomly	Local discrete	Global discrete

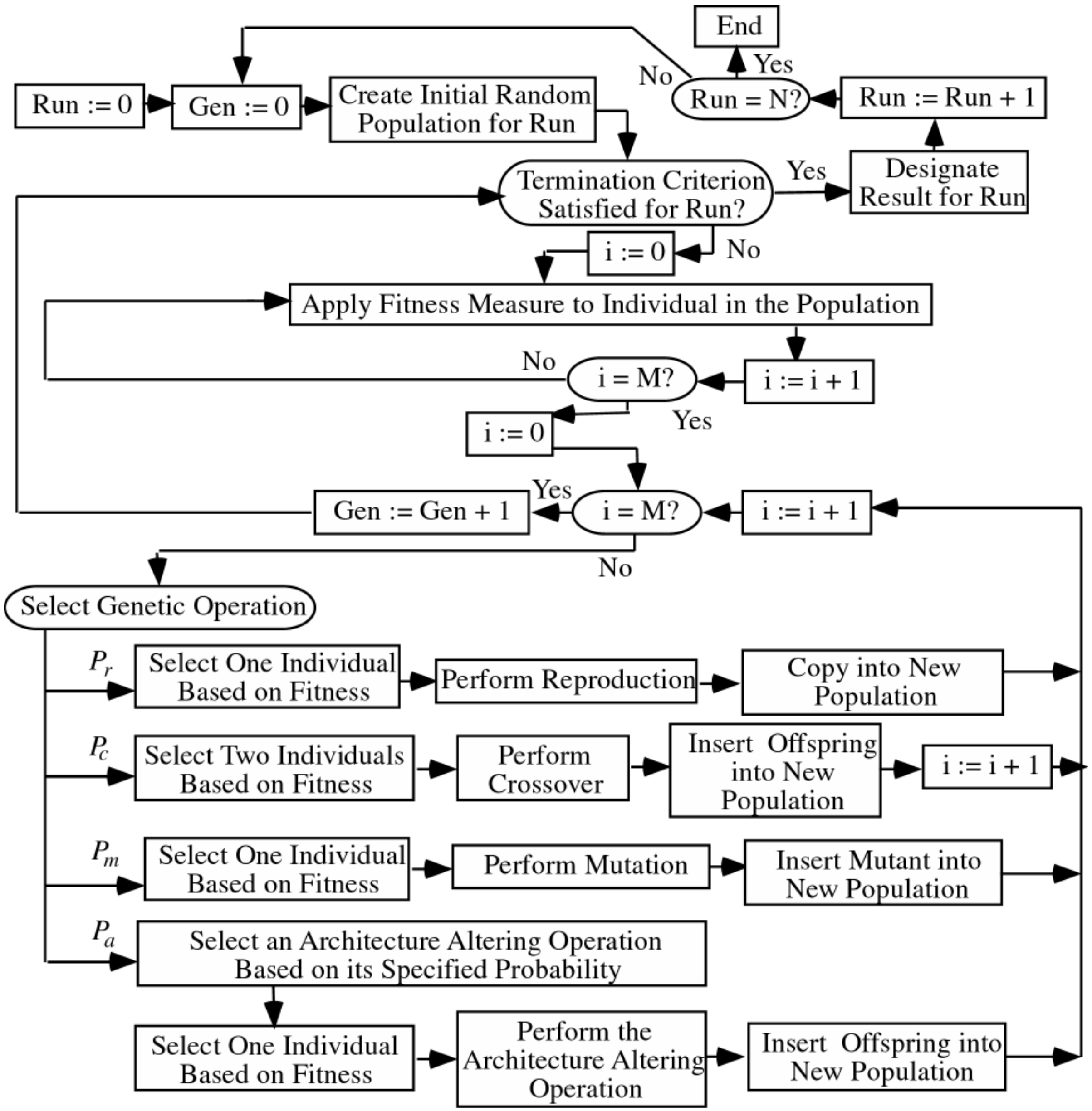
(μ+λ) Evolutionary Strategies



Estrategias Evolutivas
Tipo (μ, λ)



Genetic Programming



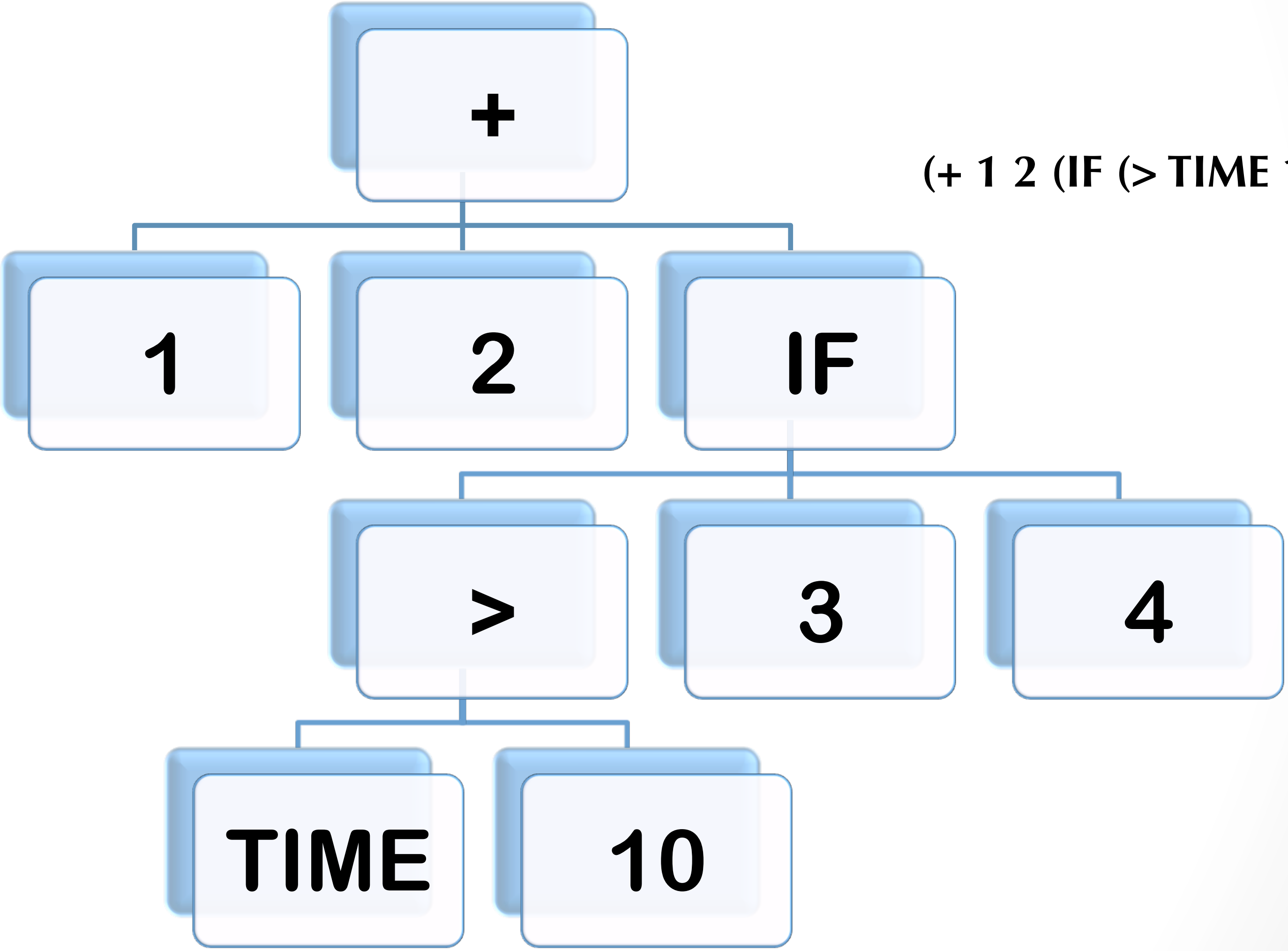
GP'S Flow Diagram

C program

```
int foo (int time)
{
    int temp1, temp2;
    if (time > 10)
        temp1 = 3;
    else
        temp1 = 4;
    temp2 = temp1 + 1 + 2;
    return (temp2);
}
```

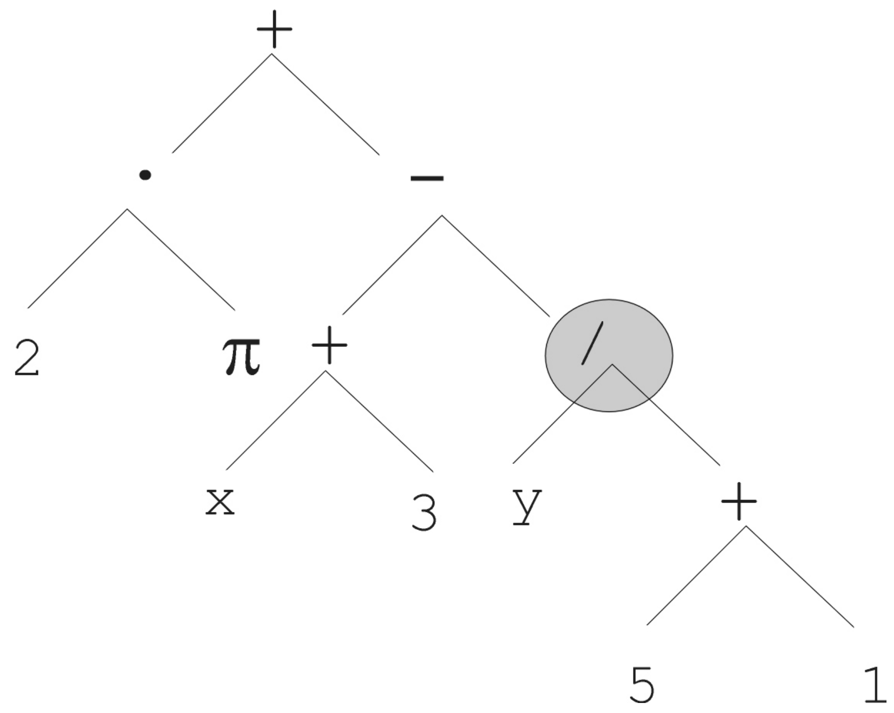
time	result
0	6
1	6
2	6
3	6
4	6
5	6
6	6
7	6
8	6
9	6
10	6
11	7
12	7

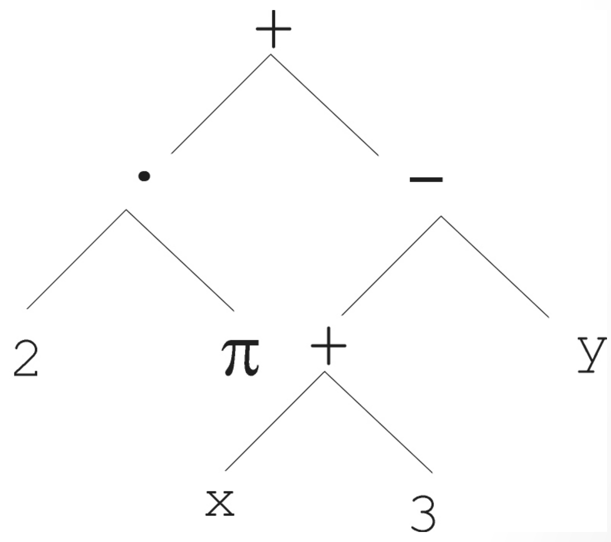
Tree representation



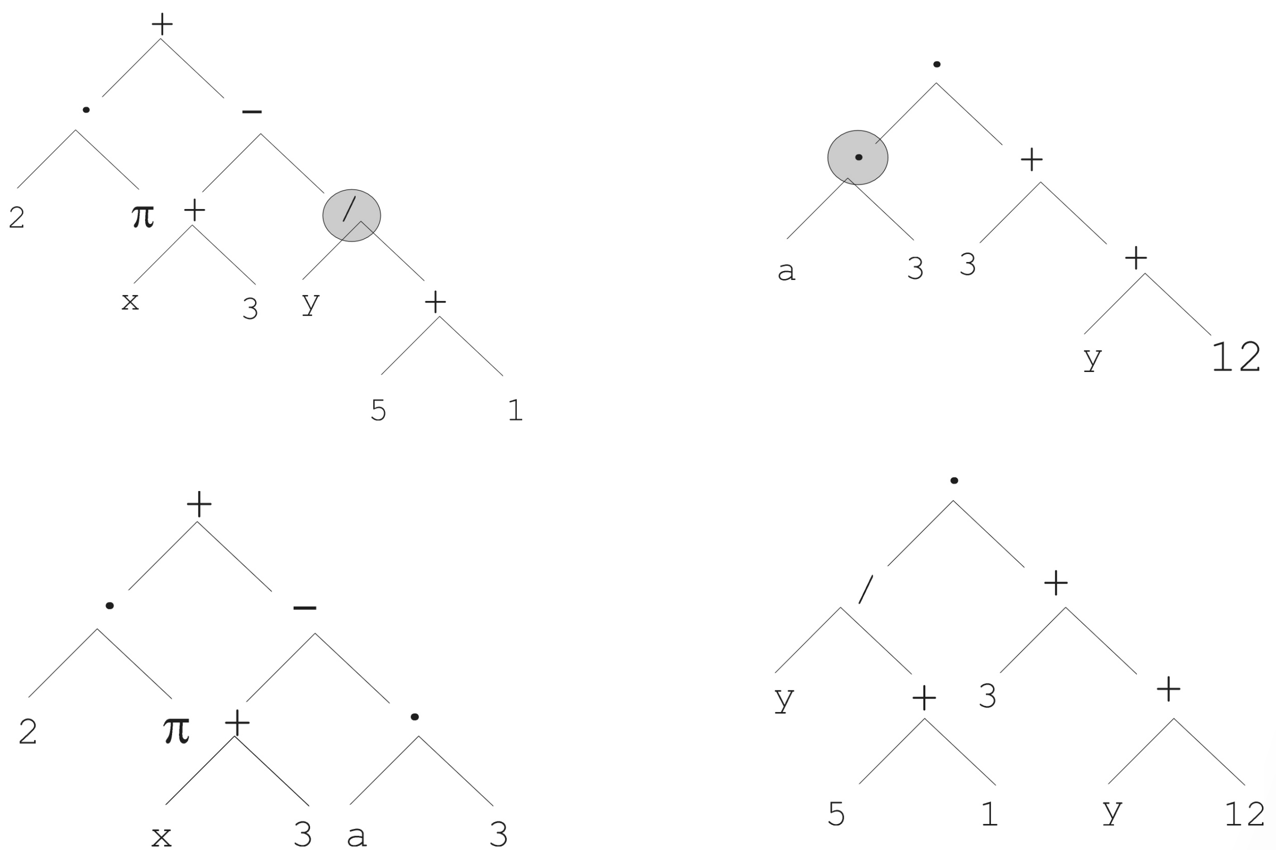
(+ 1 2 (IF (> TIME 10) 3 4))

Mutation example





Crossover

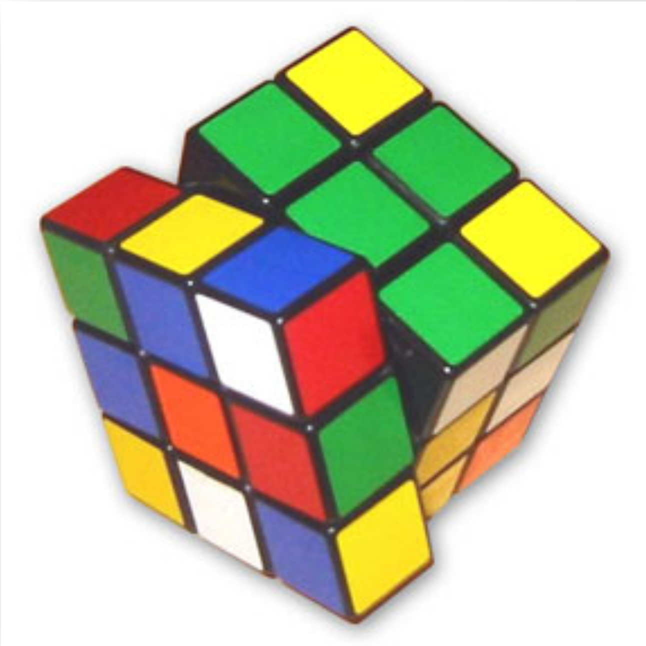


Fitness

- How to measure the quality of a problem?
  - Number of errors, impact of the errors, computing time, computational complexity, etc.
- Bloating.

Our experience with GP

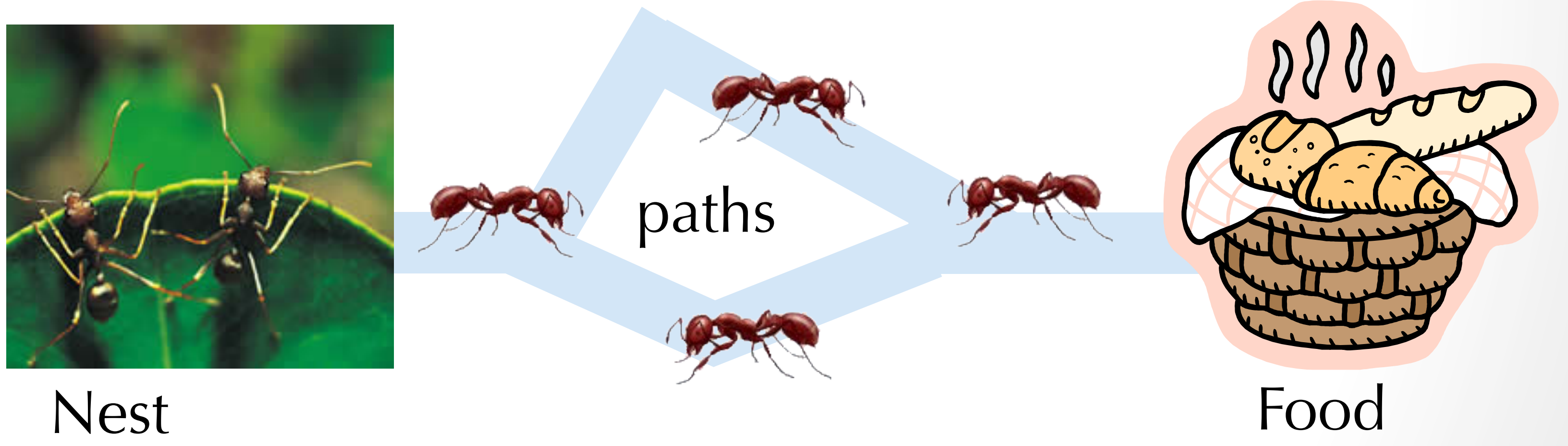
- I directed an undergraduate thesis on GP.
- UC3M GECCO 2009 GP Rubik's cube team.
- We were the only participants!



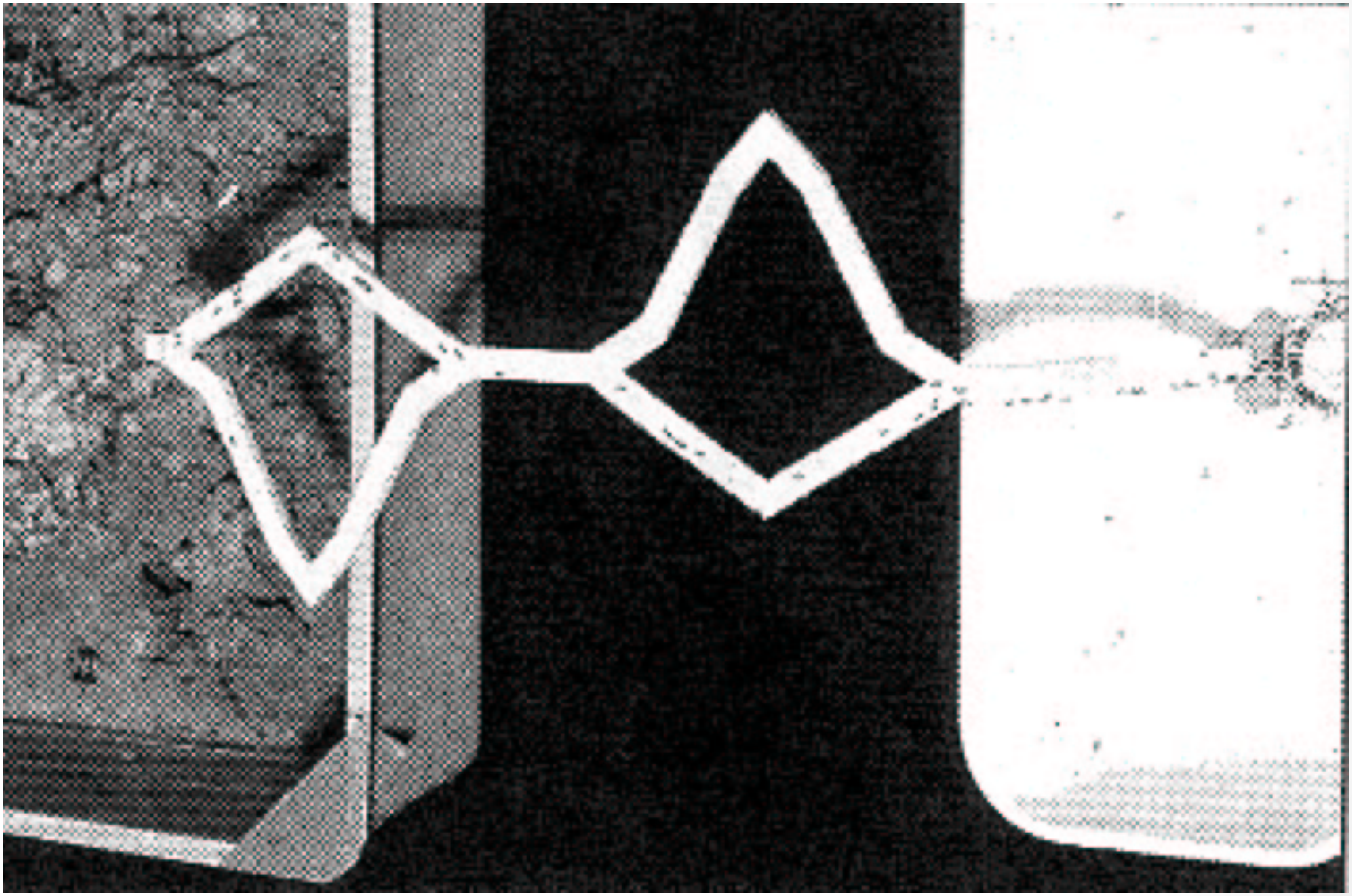
Colonia de Hormigas

More inspiration on nature

- Ant colony optimization.



Ant colony in action



More formally

- For a connected graph G=(N,A) the ant colony find the shortest path between two nodes.
- There is an "artificial pheromone footprint" associated with every arc in A.
- Ants can "read" and "write" that footprint.
- Highly transited arcs have a higher footprint.

Final Remarks

Final remarks

- Differential evolution.
- Estimation of distribution algorithms.
- Particle swarms.
- These approaches have seen many important practical results.
- Inspiration from nature does not stops here!

Homework!

- Read:
  - Von Zuben, Fernando J. "Computação evolutiva: uma abordagem pragmática."
- Start getting familiarized with IPython, numpy, scipy, scikit.learn, inspyred and DEAP.