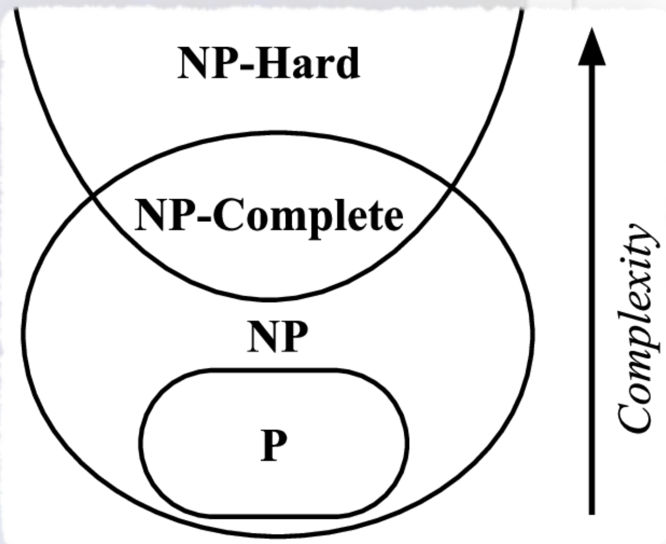
Heuristics & Metaheuristics for Optimization & Learning

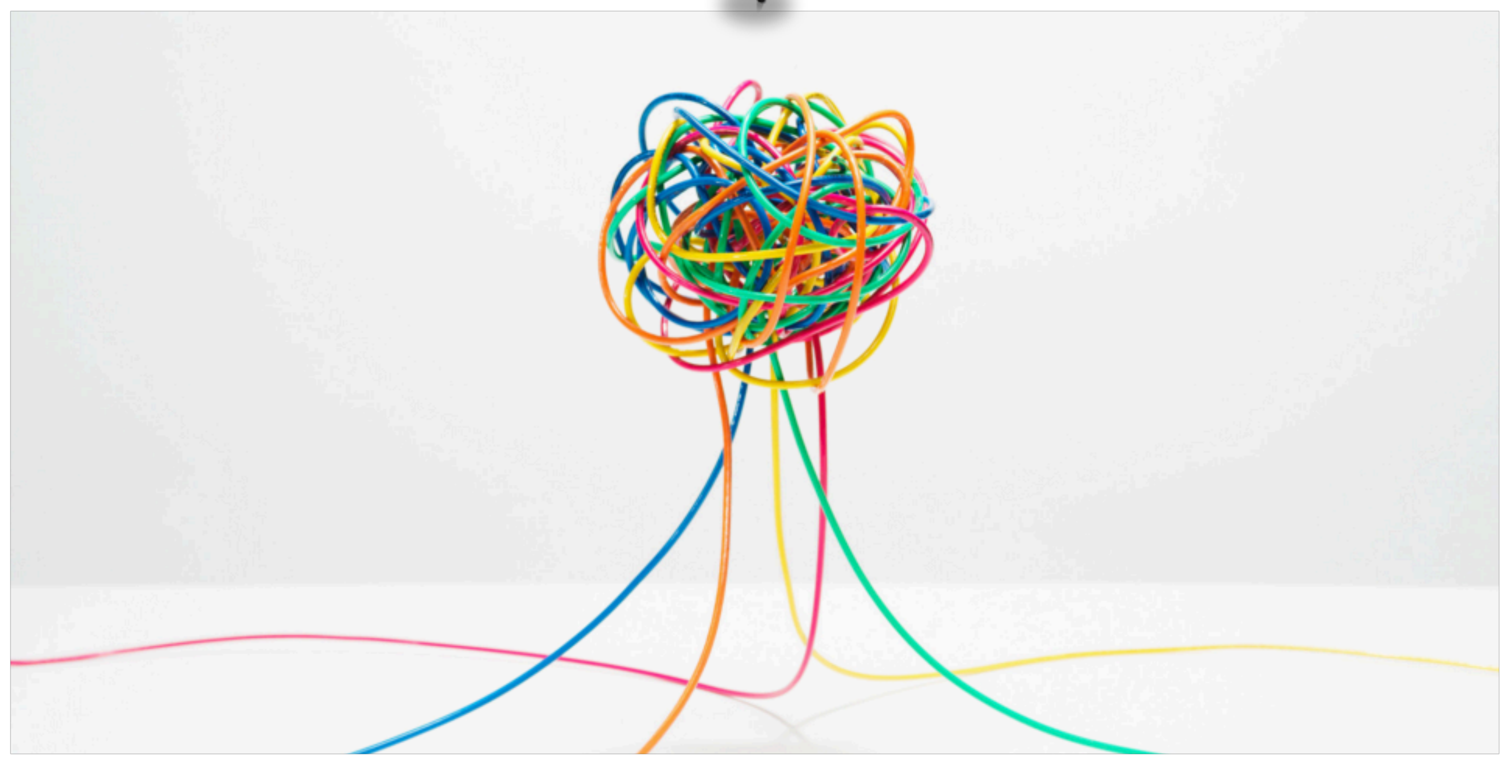
Mario Pavone

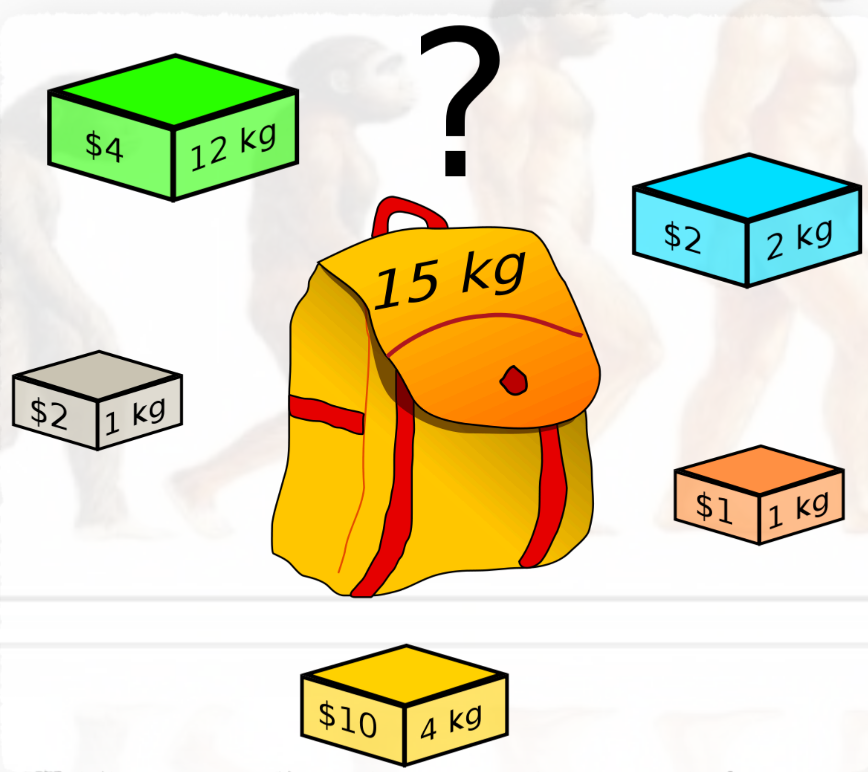
mpavone@dmi.unict.it
http://www.dmi.unict.it/mpavone/

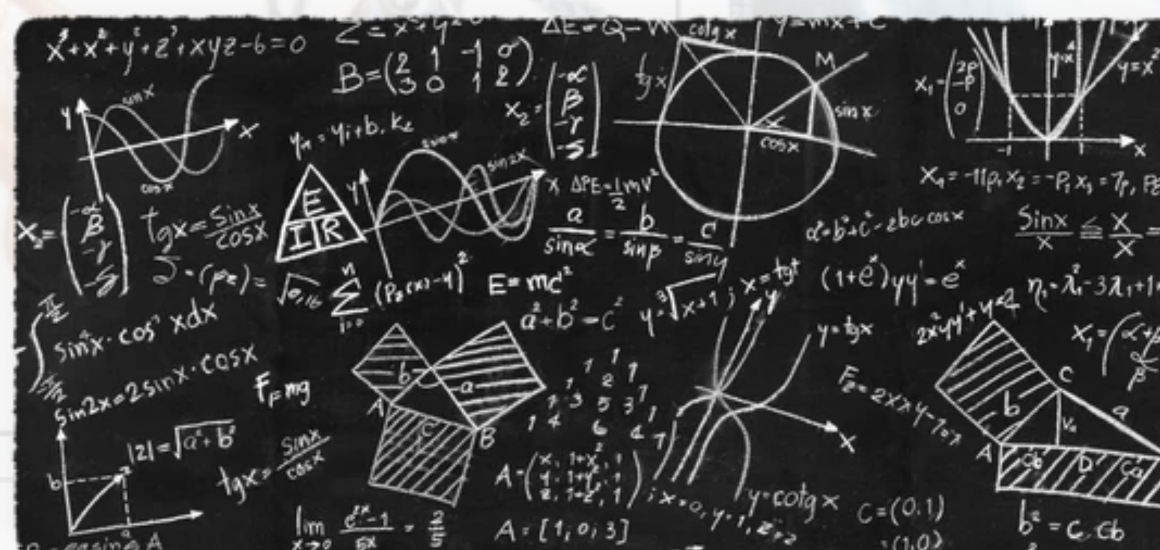
AA. 2024/2025 - LM 18

Complex Problems: NP-complete

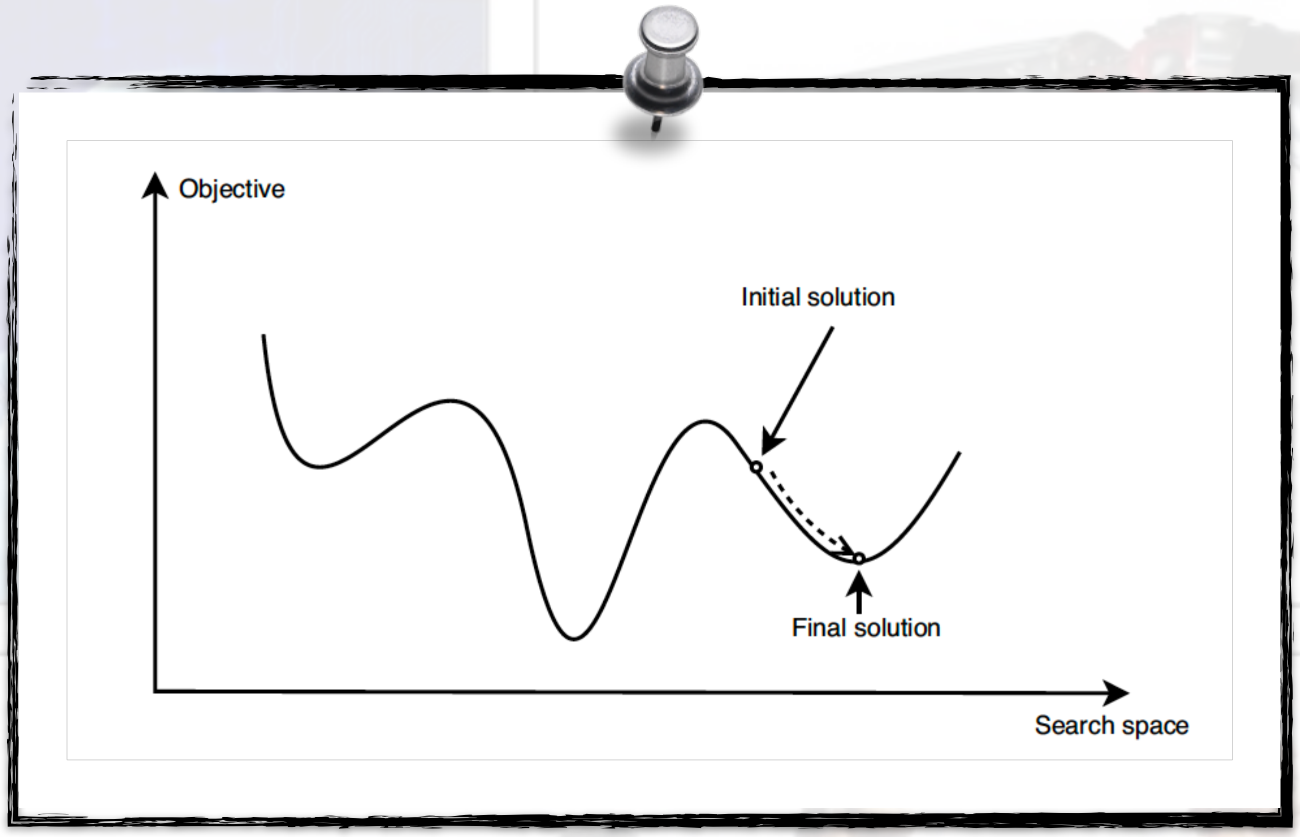


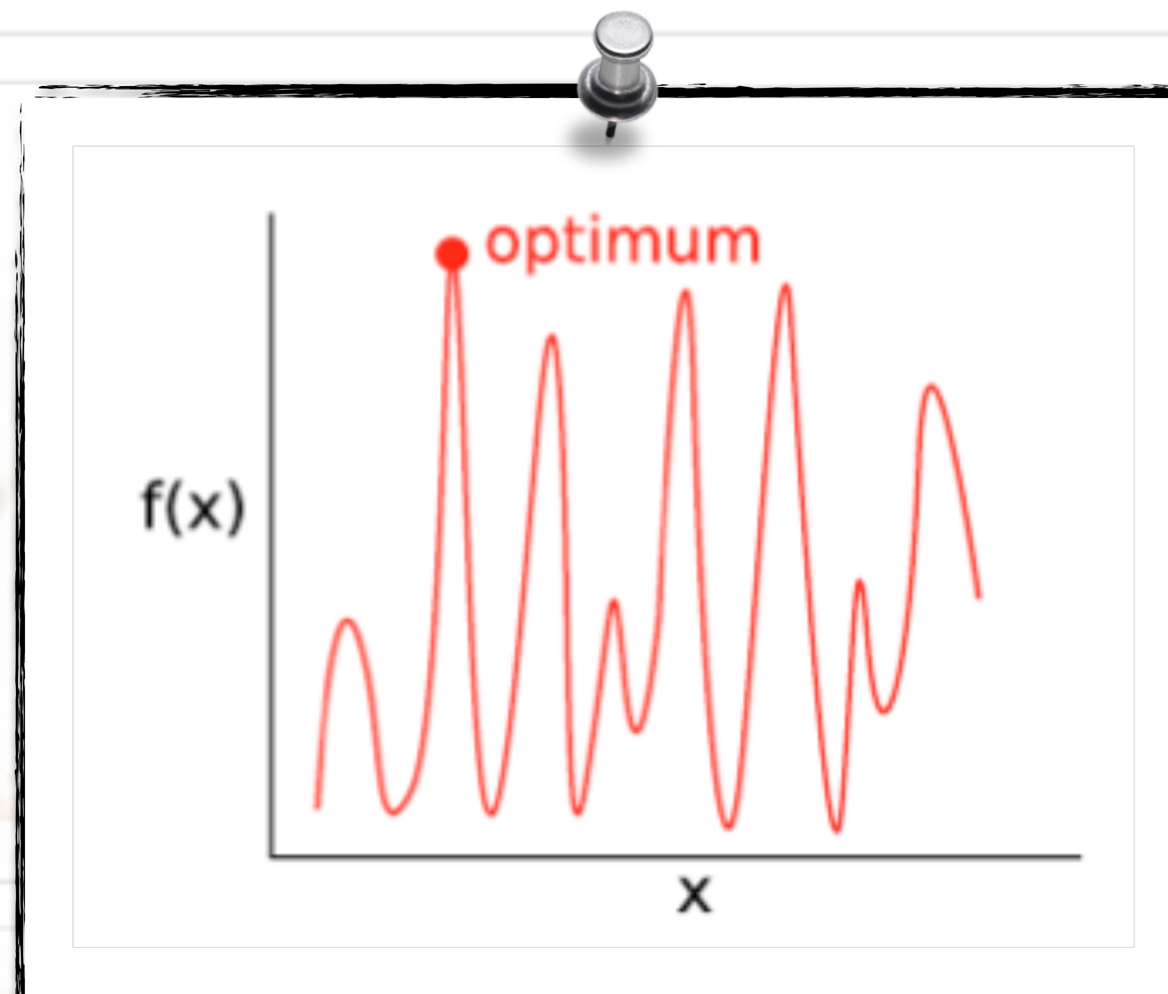






Complex Problems: Local Optima





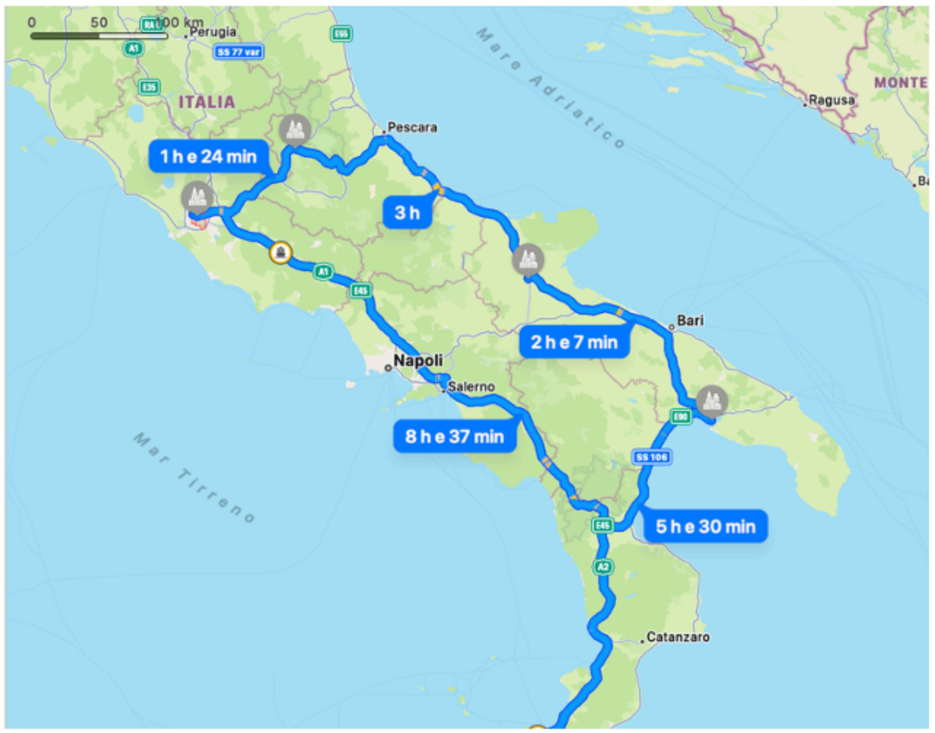
Success Key: how the search space is visited

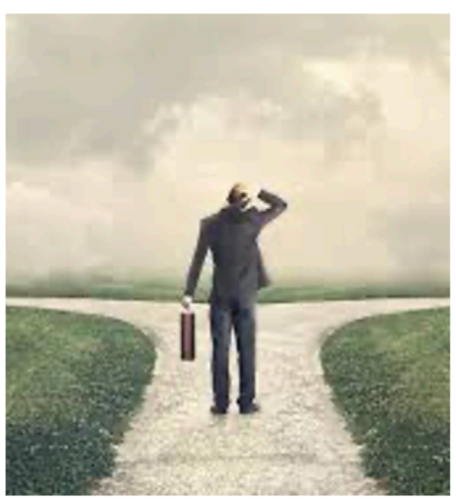
Energy level	No. of Conformations
0	36.098.079
-1	31.656.934
-2	12.473.446
-3	2.934.974
-4	517.984
-5	77.080
-6	10.364
-7	1194
-8	96
-9	**4**
Total	**83.779.155**

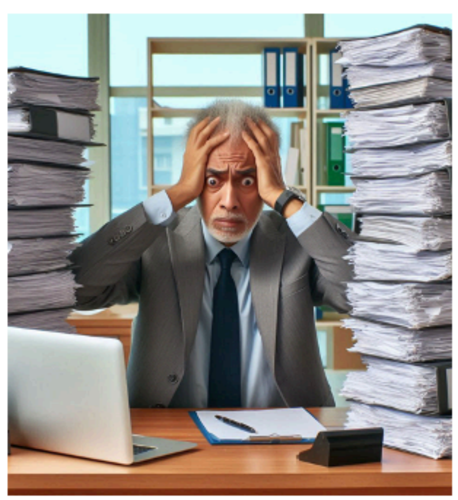
Uncertain & Dynamic Scenarios

Time Constraints & Big Data

Time







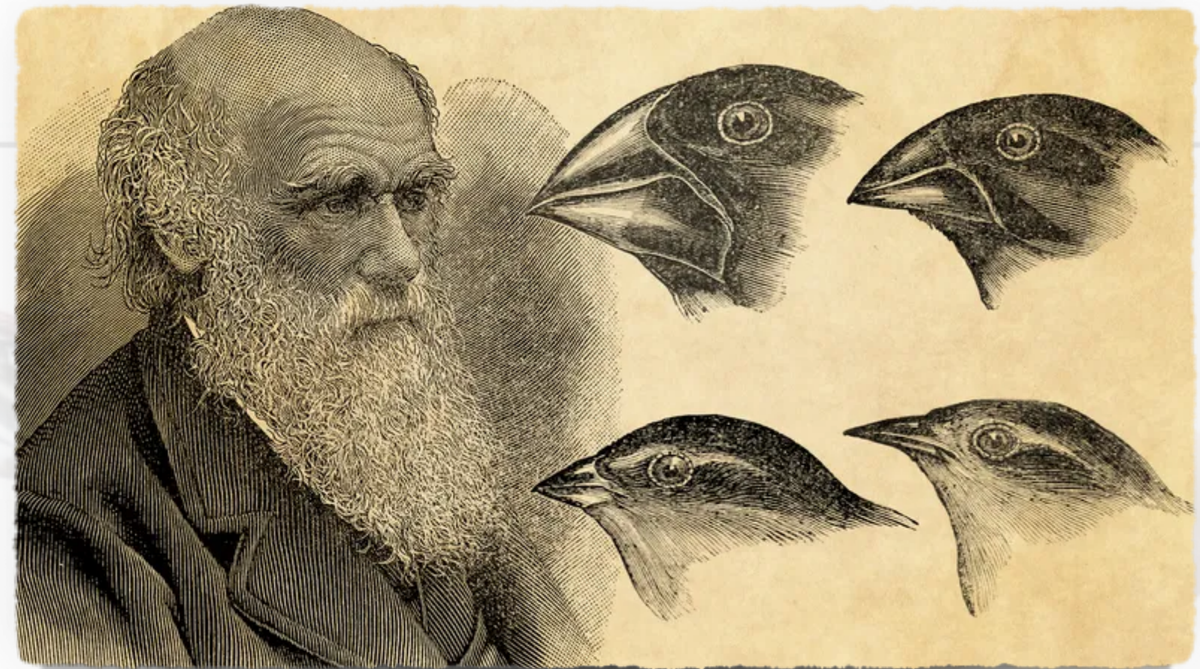
Uncertainty

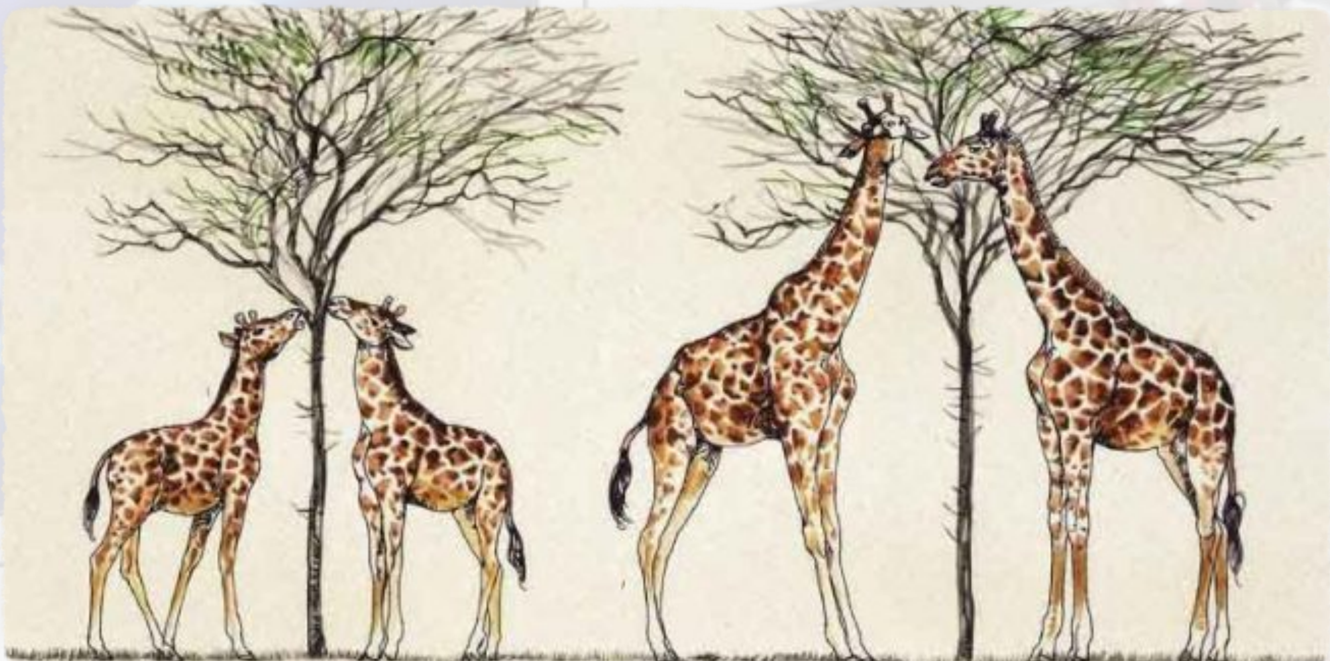
Big Data


E ADESSO
COSA
FACCIO?

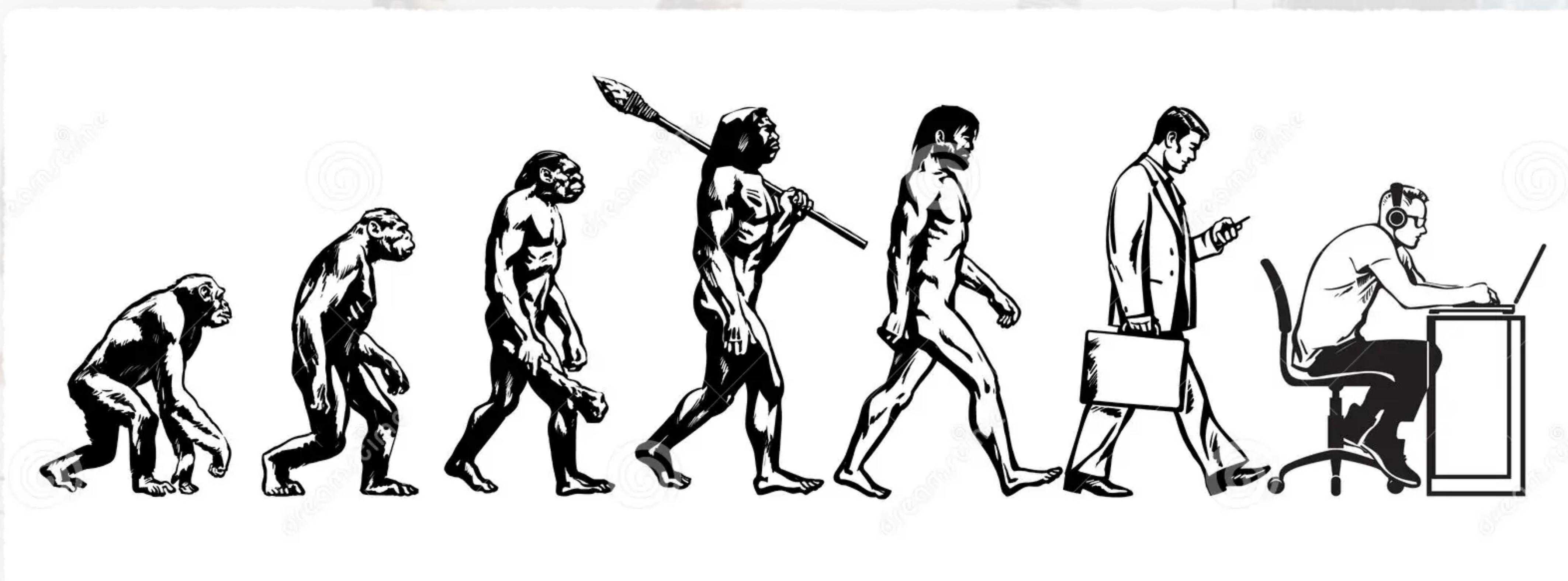
Look the nature and be inspired!


HOW

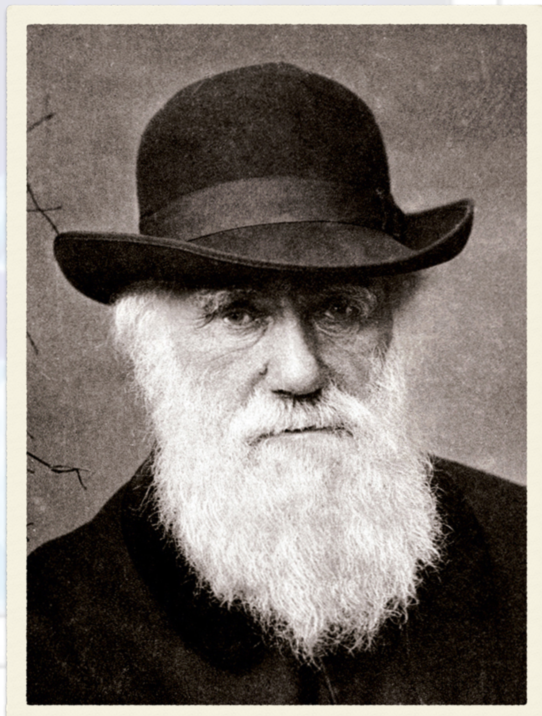




Teoria dell'Evoluzione



Teoria dell'Evoluzione



Selezione Naturale del *migliore*

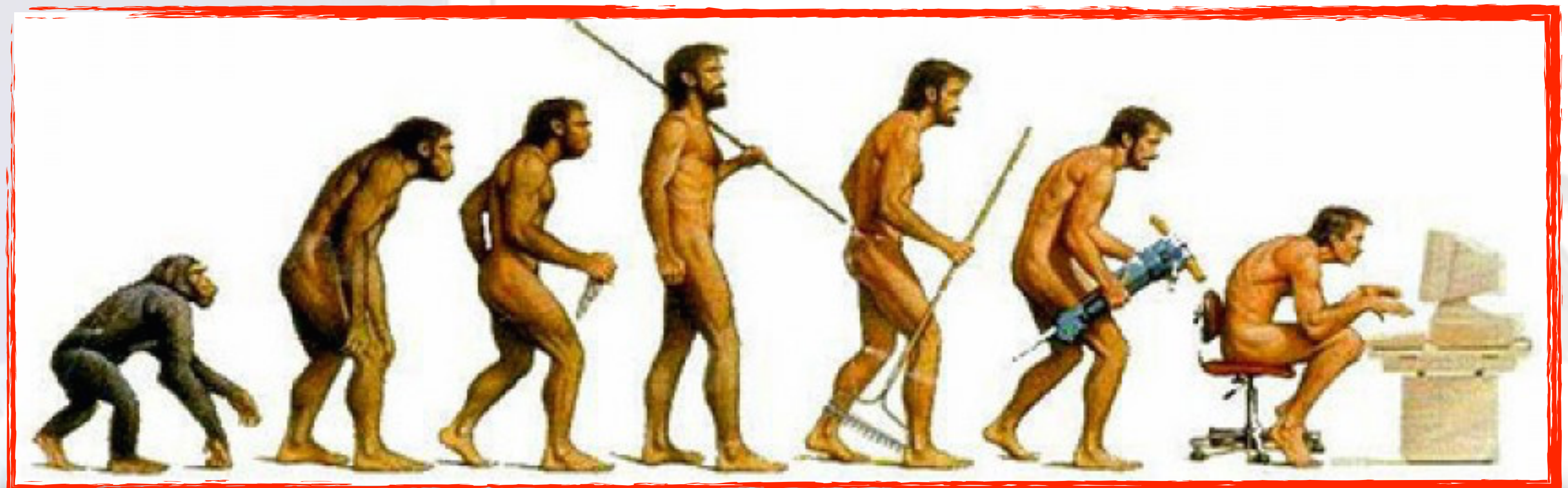
Adattamento

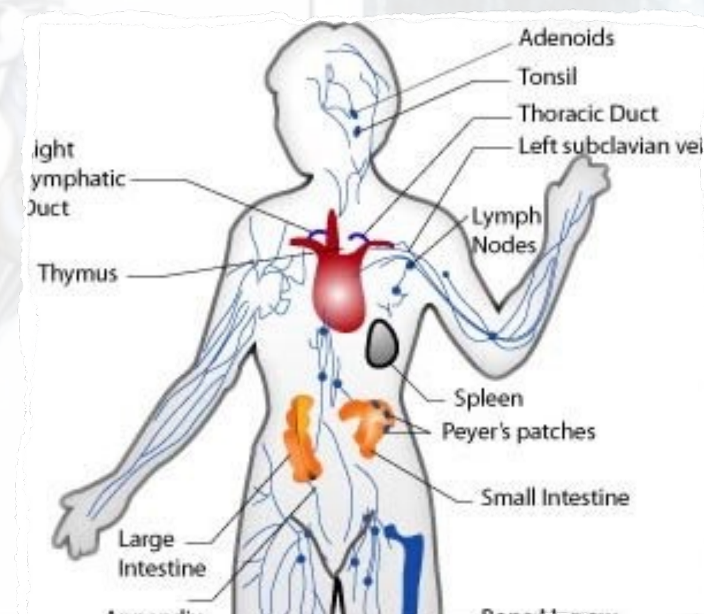
Crossover

Mutazione

Evoluzione

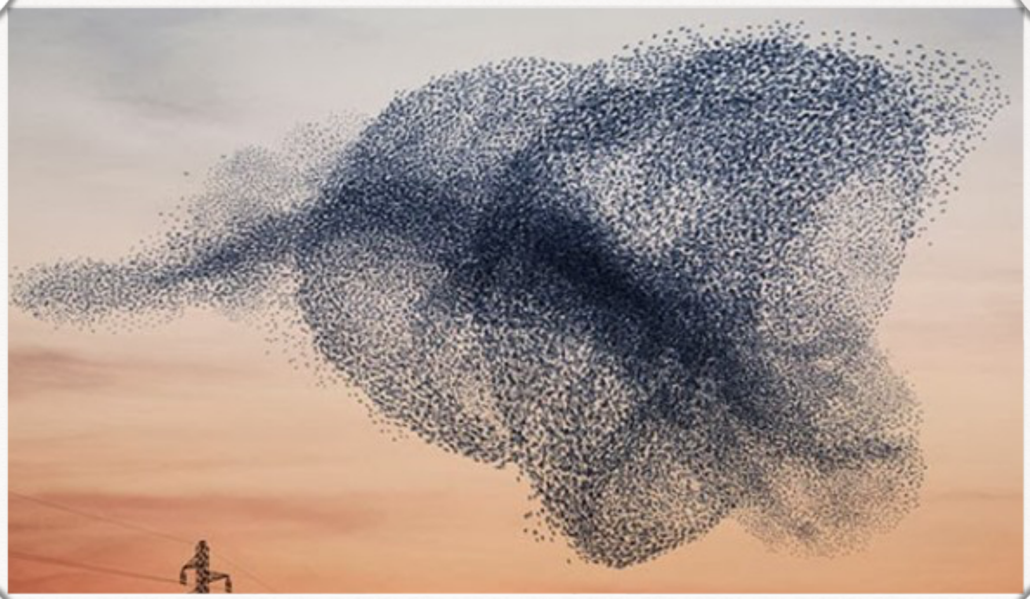
Algoritmi Bio- & Nature-Inspired

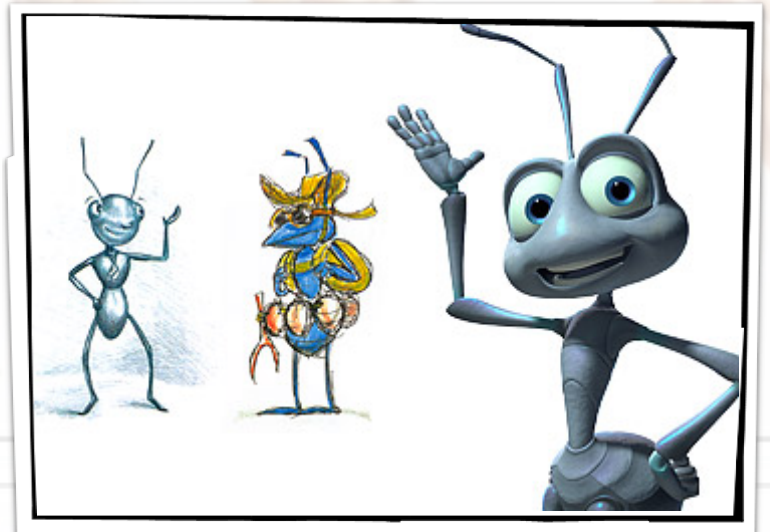


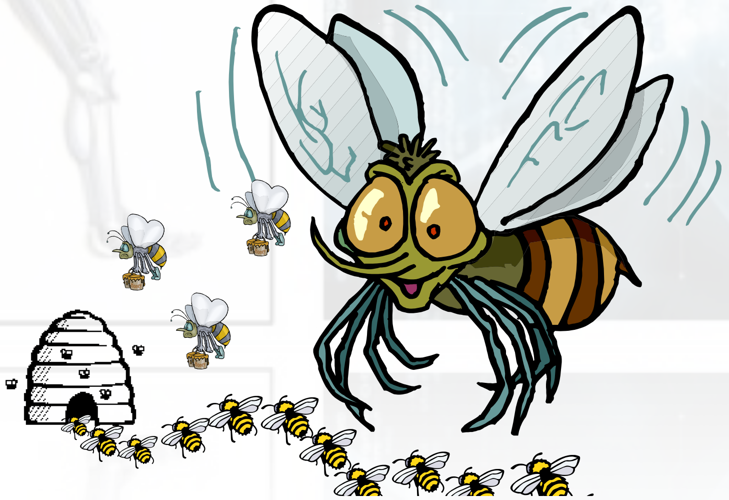


Swarm Intelligence

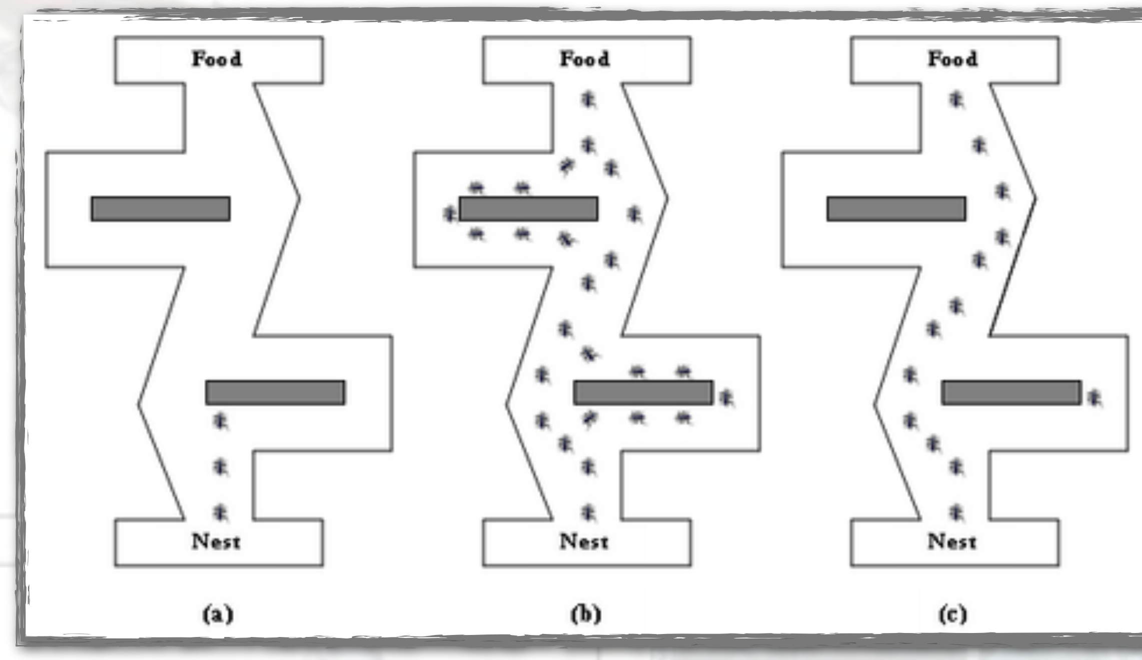
Comportamento Collettivo, Decentralizzato & Self-Adaptive

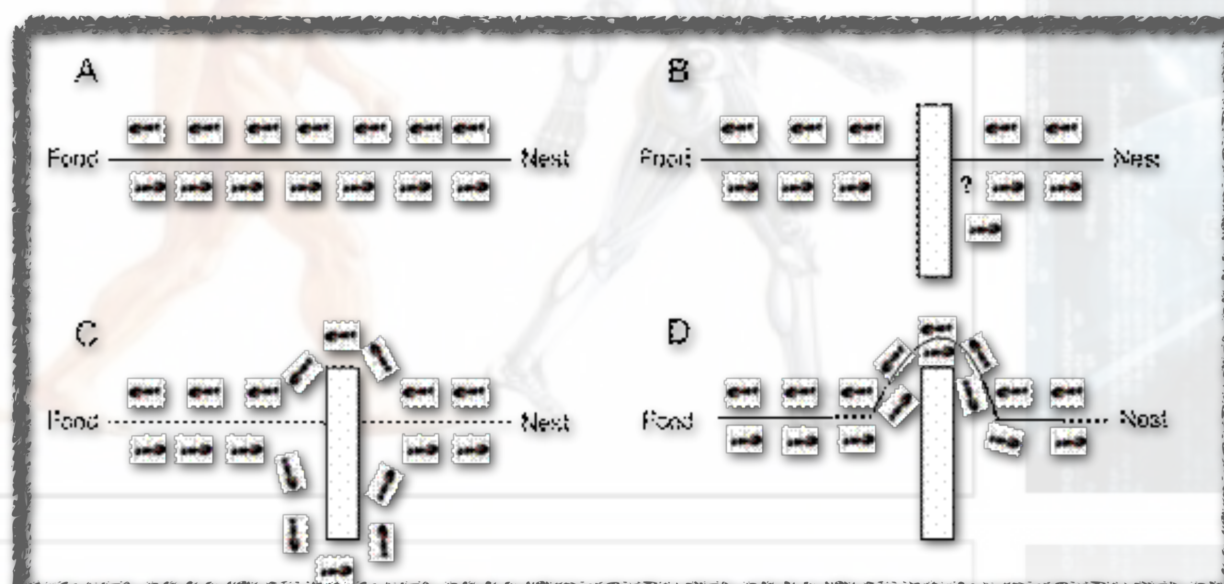




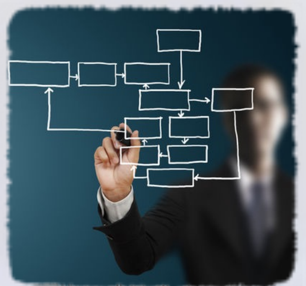


Colonie di Formiche



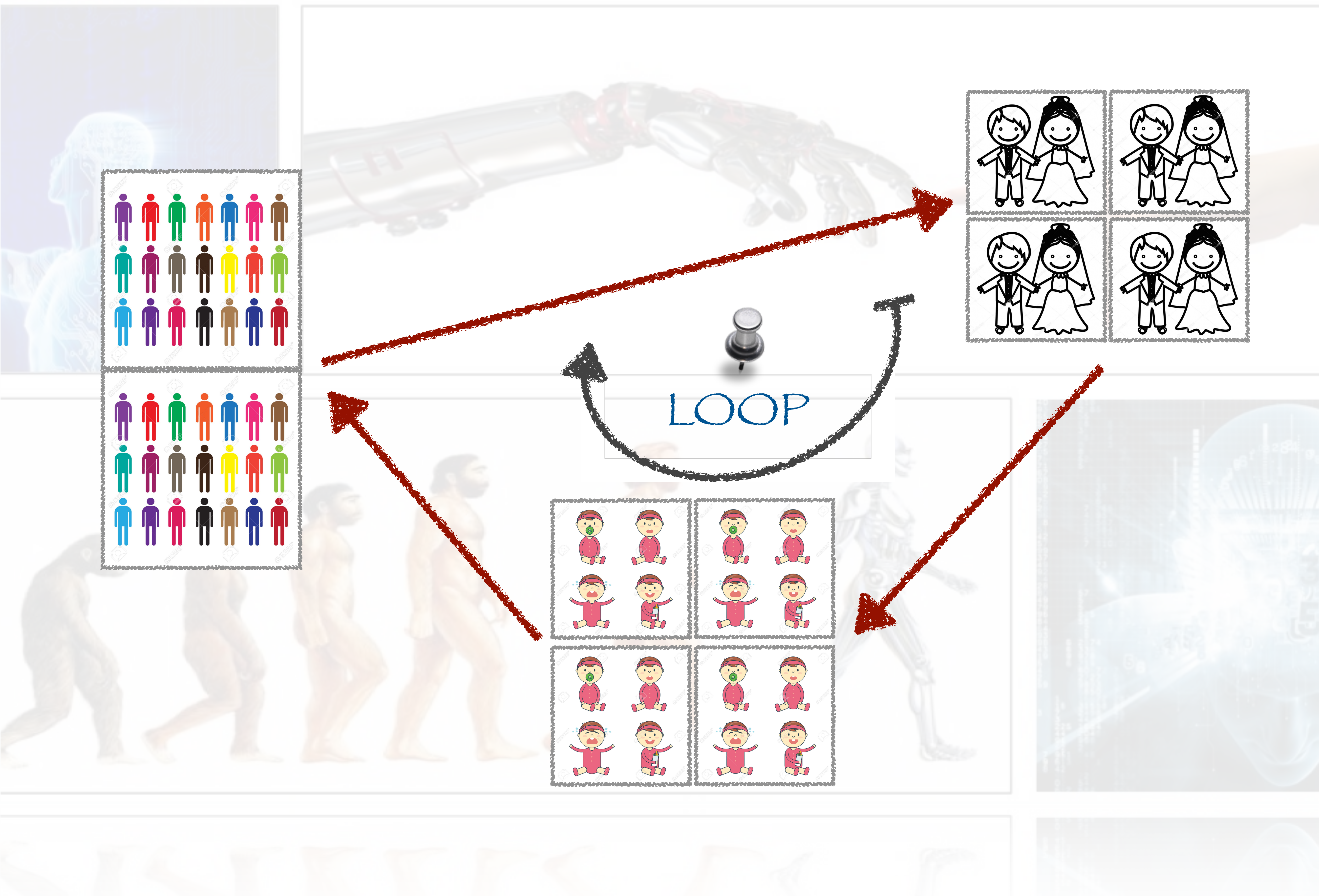


Organization & Cooperation

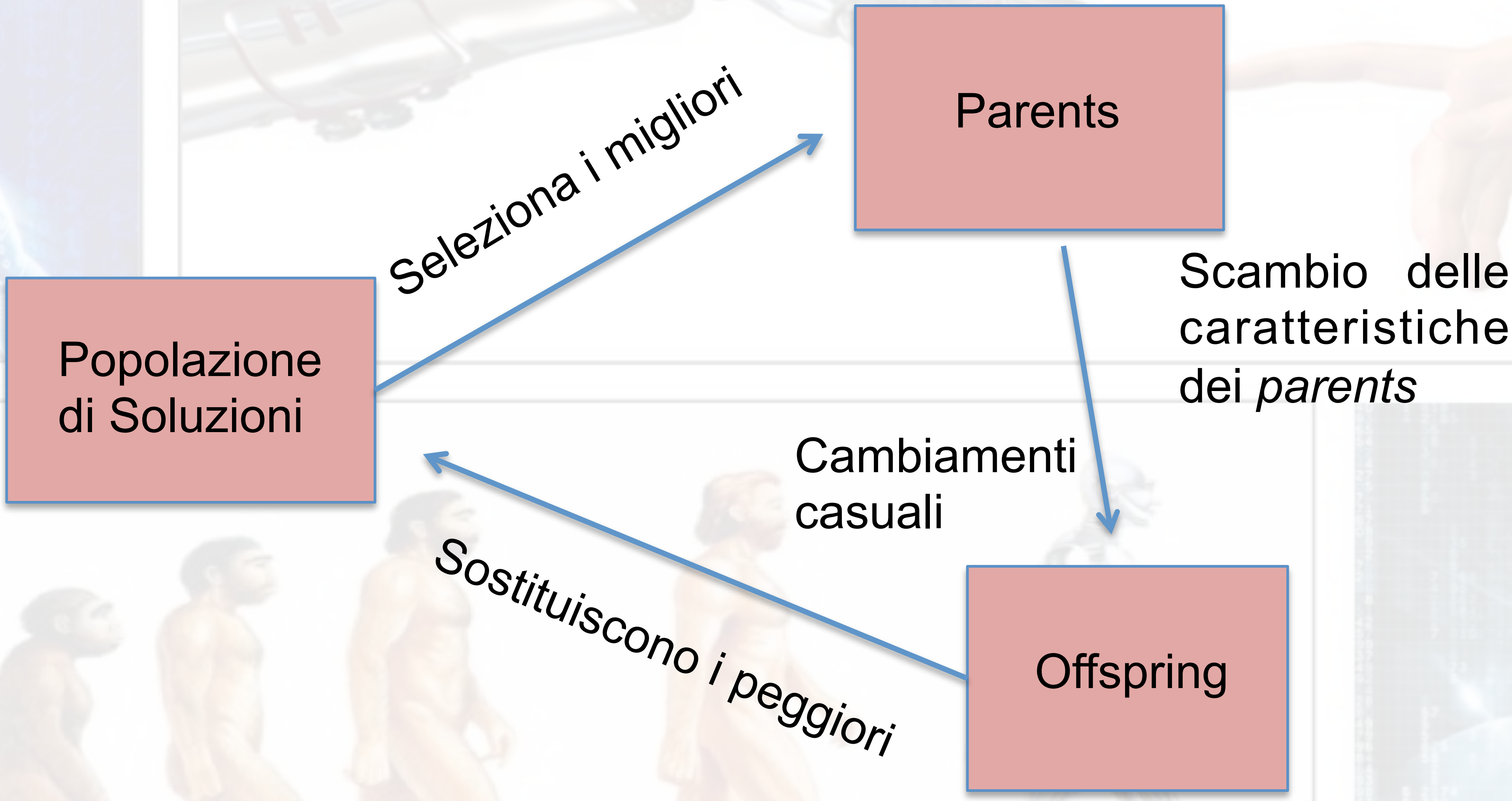


Come funzionano?

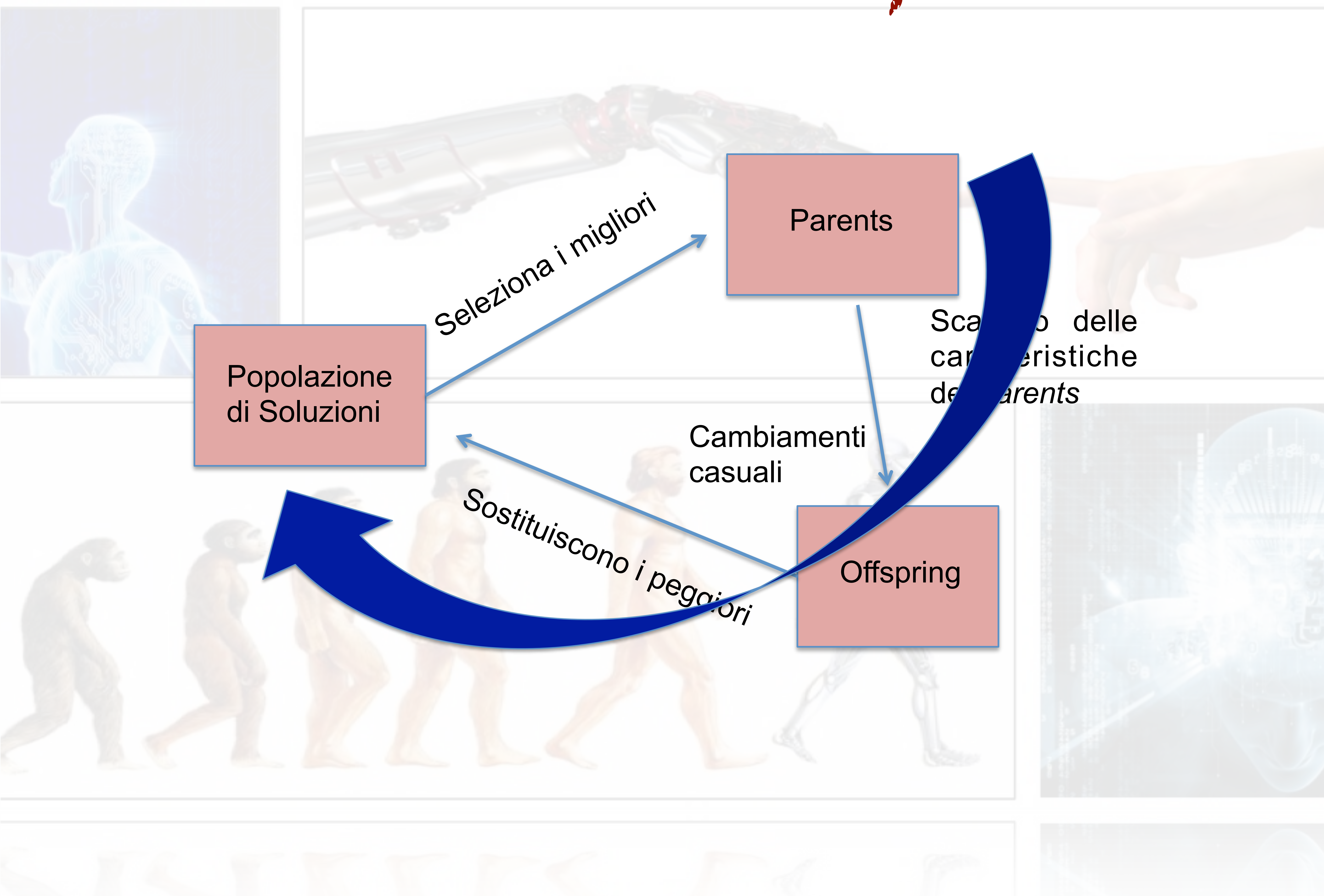
Evoluzione Naturale



Evoluzione Artificiale



Evoluzione Artificiale



Ma Perché Sviluppare una nuova classe di Algoritmi se già esistono i Metodi Classici?!?!

Confronti

[E.G. Talbi, "Metaheuristics: from Design to Implementation", Wiley & Sons, 2009]

Deterministic versus stochastic: A deterministic metaheuristic solves an optimization problem by making deterministic decisions (e.g., local search, tabu search). In stochastic metaheuristics, some random rules are applied during the search (e.g., simulated annealing, evolutionary algorithms). In deterministic algorithms, using the same initial solution will lead to the same final solution, whereas in stochastic metaheuristics, different final solutions may be obtained from the same initial solution. This characteristic must be taken into account in the performance evaluation of metaheuristic algorithms.

Iterative versus greedy: In iterative algorithms, we start with a complete solution (or population of solutions) and transform it at each iteration using some search operators. Greedy algorithms start from an empty solution, and at each step a decision variable of the problem is assigned until a complete solution is obtained. Most of the metaheuristics are iterative algorithms.

Population-based search versus single-solution based search: Single-solution based algorithms (e.g., local search, simulated annealing) manipulate and transform a single solution during the search while in population-based algorithms (e.g., particle swarm, evolutionary algorithms) a whole population of solutions is evolved. These two families have complementary characteristics: single-solution based metaheuristics are exploitation oriented; they have the power to intensify the search in local regions. Population-based metaheuristics are exploration oriented; they allow a better diversification in the whole search space. In the

i Vantaggi

Molti aspetti dei *Processi Evolutivi* sono **Stocastici**

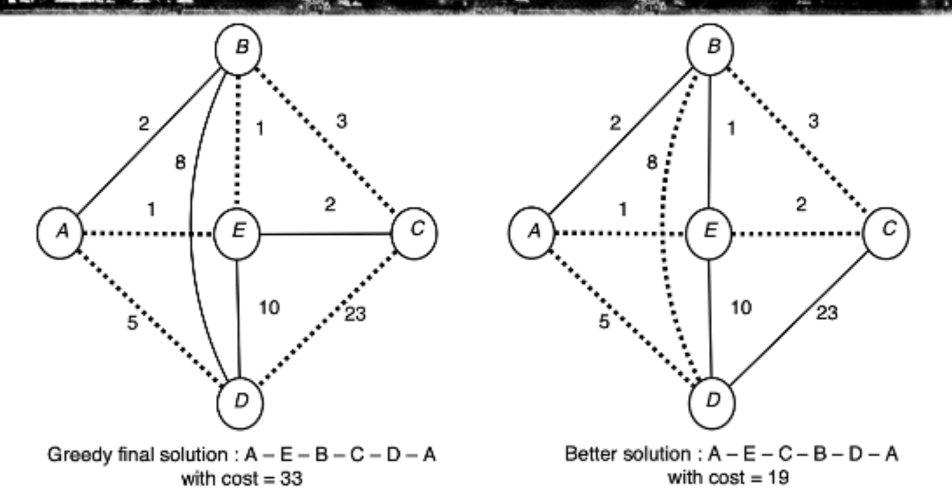


FIGURE 1.10 Illustrating a greedy algorithm for the TSP using a static heuristic. An element is associated with an edge of the graph, and the local heuristic consists in choosing the nearest neighbor. The obtained solution is $(A - E - B - C - D - A)$ with a total cost of 33, whereas a better solution with a cost of 19 is given (right).

In most of Optimization Problems, the local view of Greedy Heuristics decreases their performance compared to the Iterative Algorithms

i Vantaggi

Molti aspetti dei *Processi Evolutivi* sono **Stocastici**

Sistemi Computazionali che *non richiedono alcuna conoscenza del dominio di applicazione* (**BlackBox**)

I know it when I see it

S. Luke, *"Essentials of Metaheuristics"*, Online Version 2.2, Oct. 2015

Adatti in problemi con **incertezza**, **imprecisione** e **approssimazione**

i Vantaggi

Robustezza: in grado di affrontare ambienti con incertezza

Decentrata: senza un autorità centrale

Autonomi: possono funzionare senza l'intervento dell'utente

Flessibilità: applicabile a differenti problemi

Adattivi: in grado di gestire applicazioni in ambienti dinamici attraverso l'auto-adattamento