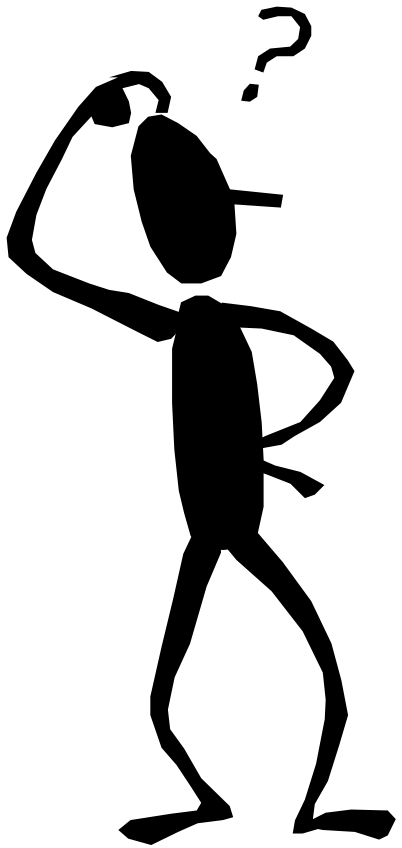
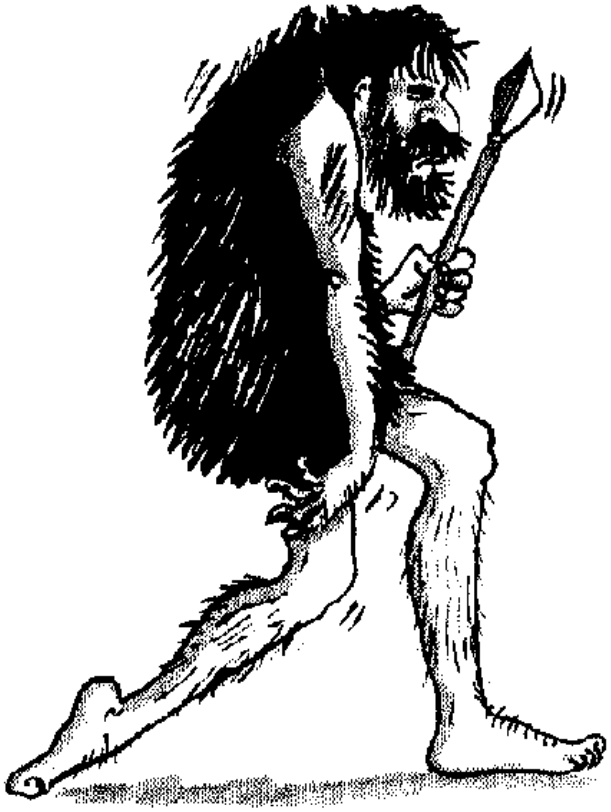
Genetic Algorithms

The Traditional Approach

- Ask an expert
- Adapt existing designs
- Trial and error

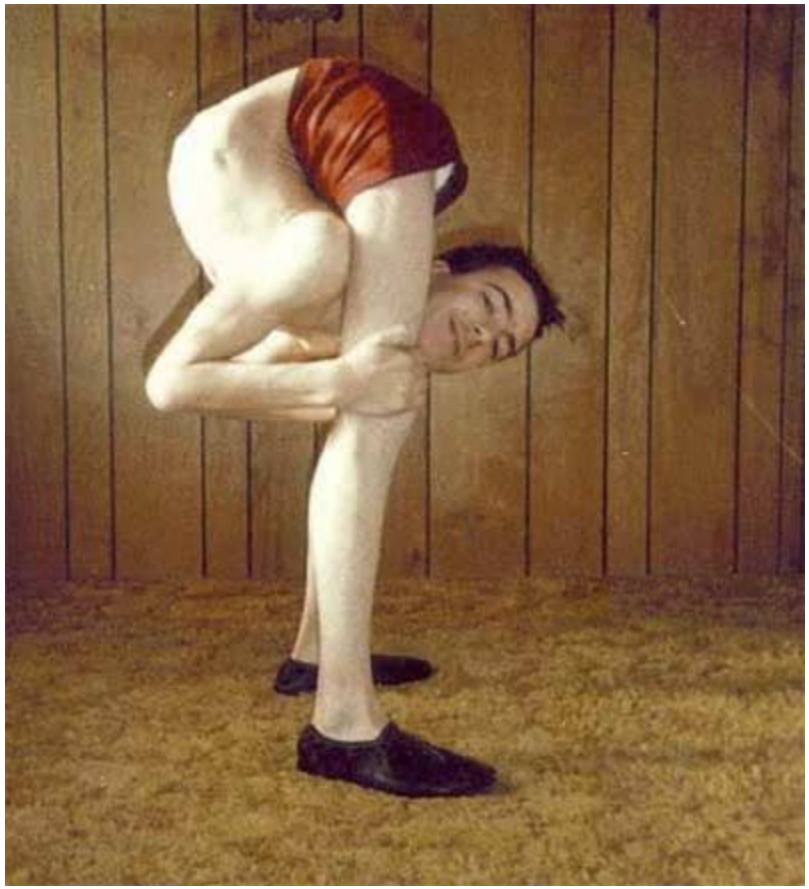


Nature's Starting Point



Alison Everitt's "A User's Guide to Men"

Optimised Man!



Example: Pursuit and Evasion

- Using NNs and Genetic algorithm
- 0 learning
- 200 tries
- 999 tries

Comparisons

- Traditional
  - best guess
    - may lead to local, not global optimum
- Nature
  - population of guesses
    - more likely to find a better solution

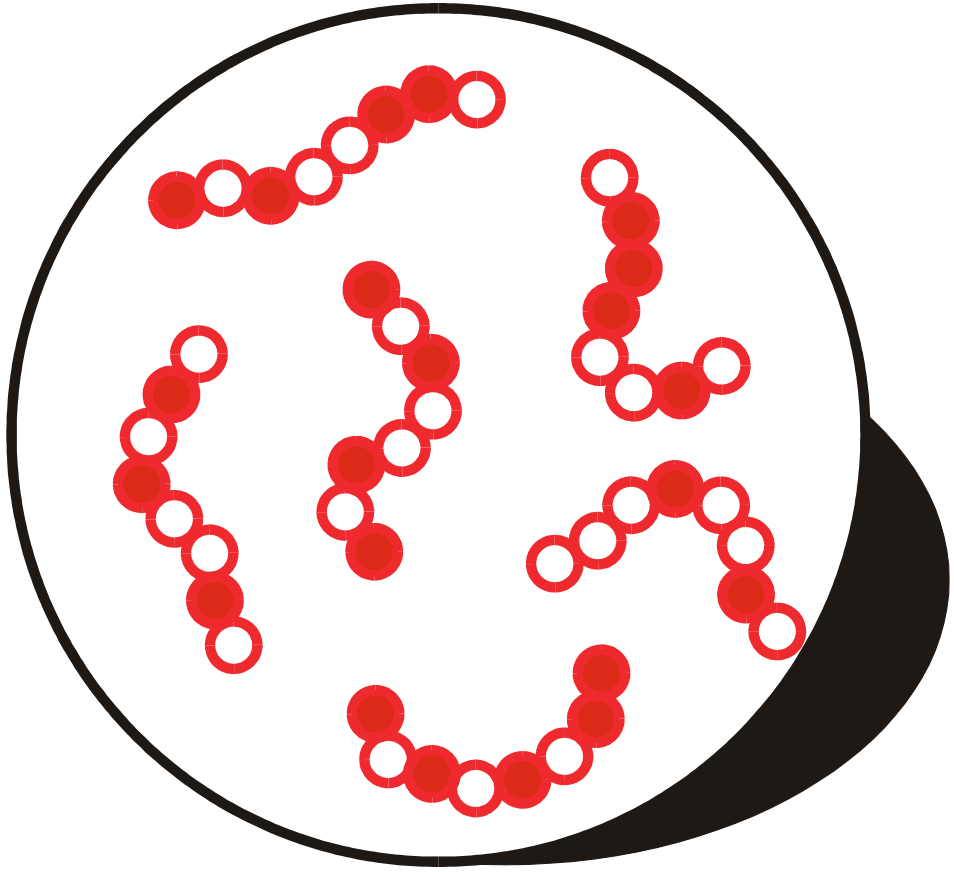
More Comparisons

- Nature
  - not very efficient
    - at least a 20 year wait between generations
    - not all mating combinations possible
- Genetic algorithm
  - efficient and fast
    - optimization complete in a matter of minutes
    - mating combinations governed only by “fitness”

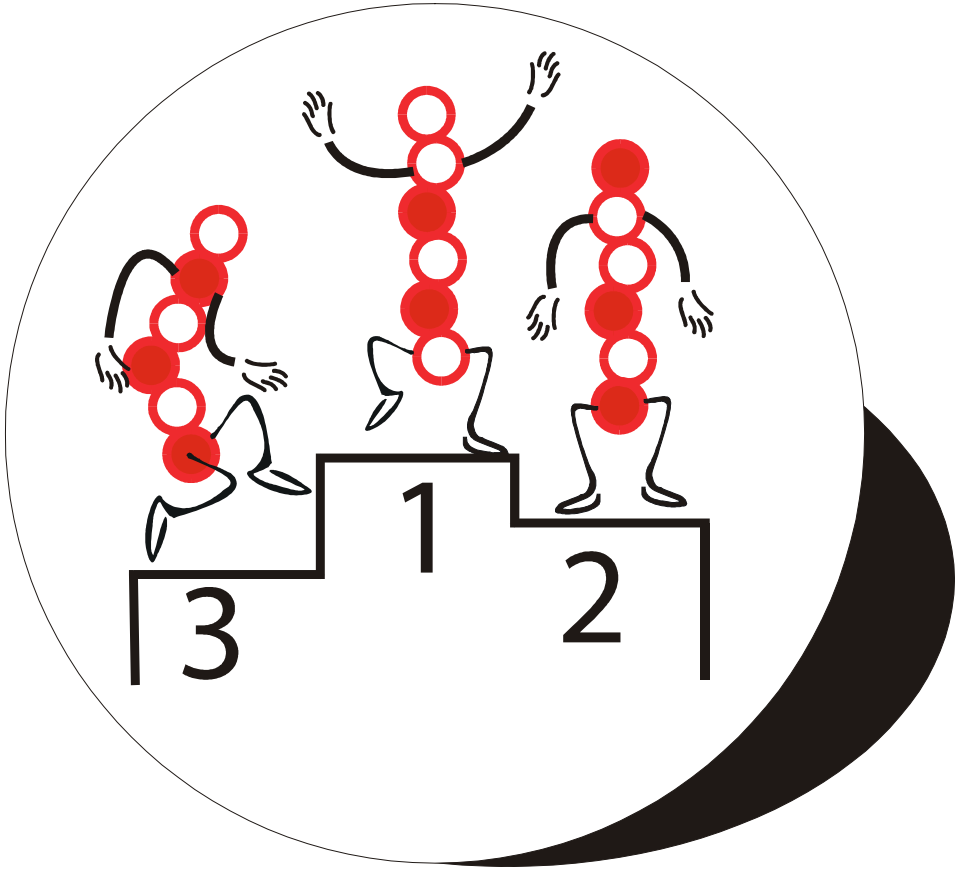
The Genetic Algorithm Approach

- Define limits of variable parameters
- Generate a random population of designs
- Assess “fitness” of designs
- Mate selection
- Crossover
- Mutation
- Reassess fitness of new population

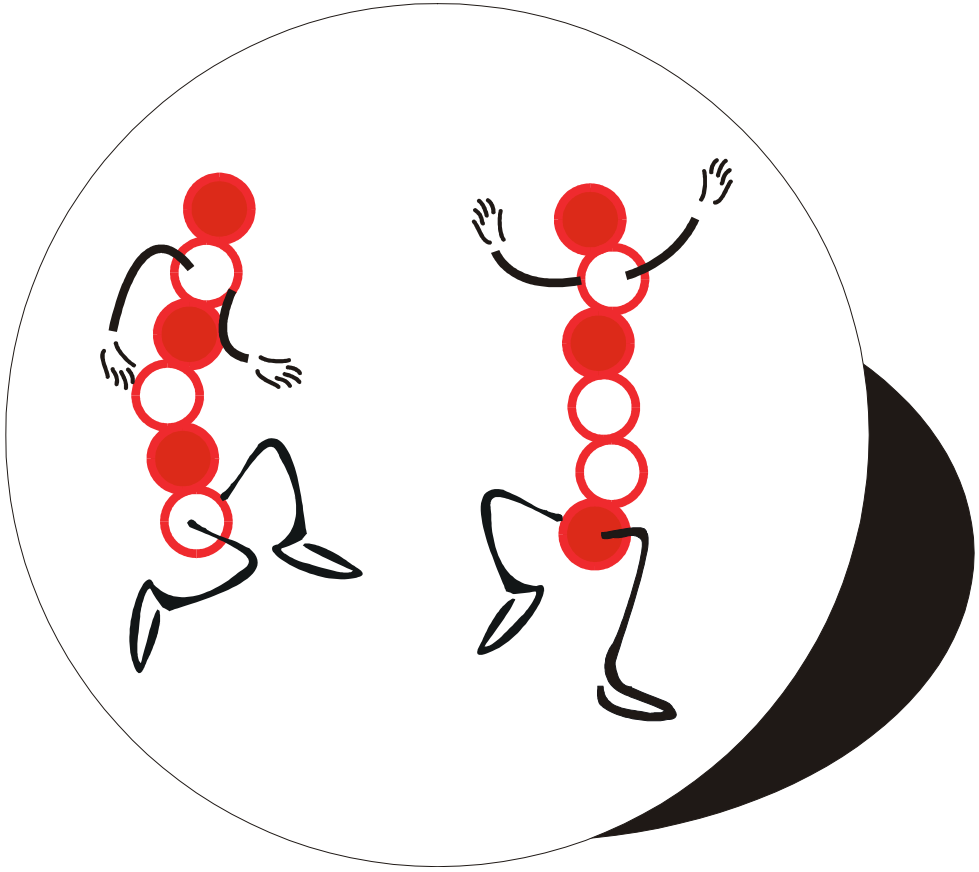
A “Population”



Ranking by Fitness:

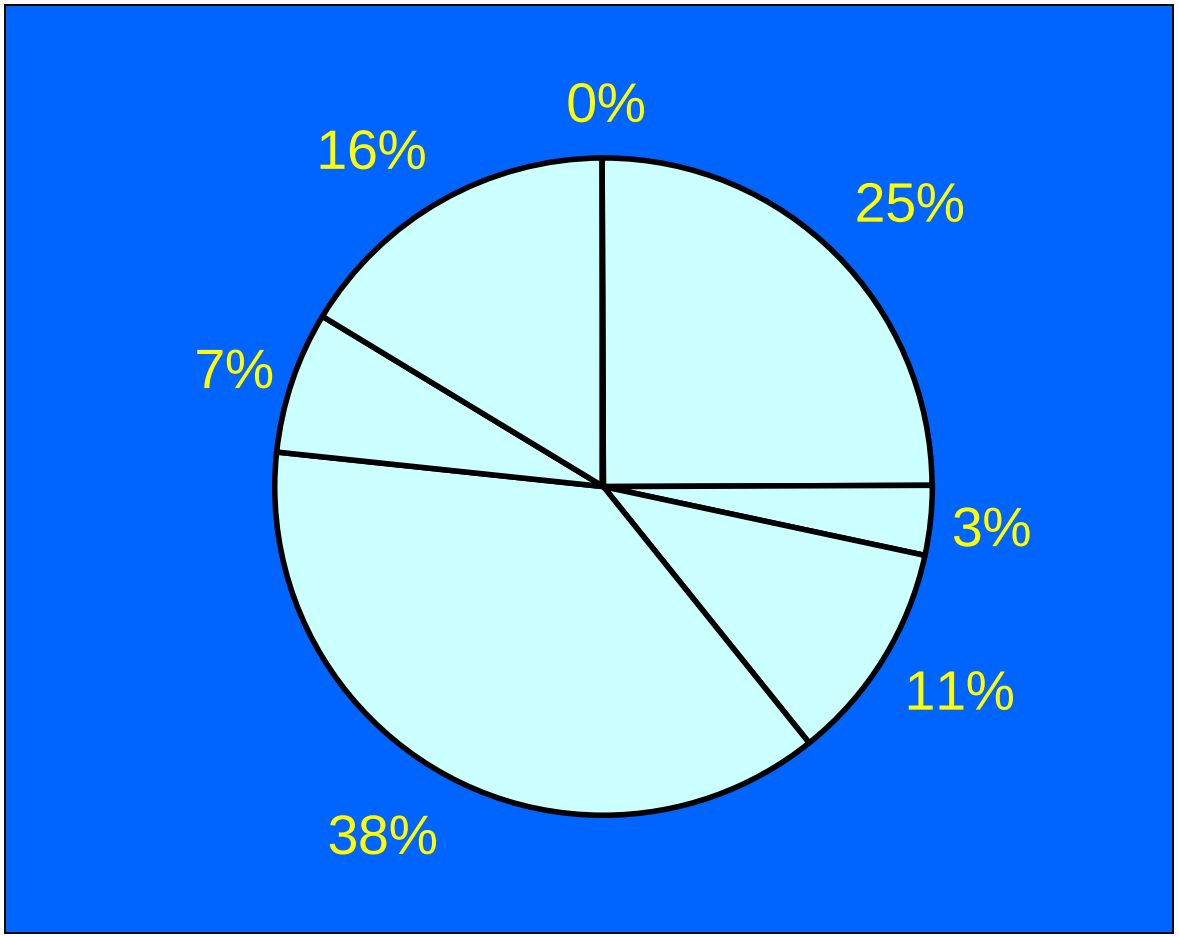


Mate Selection:
Fittest are copied and replaced less-fit



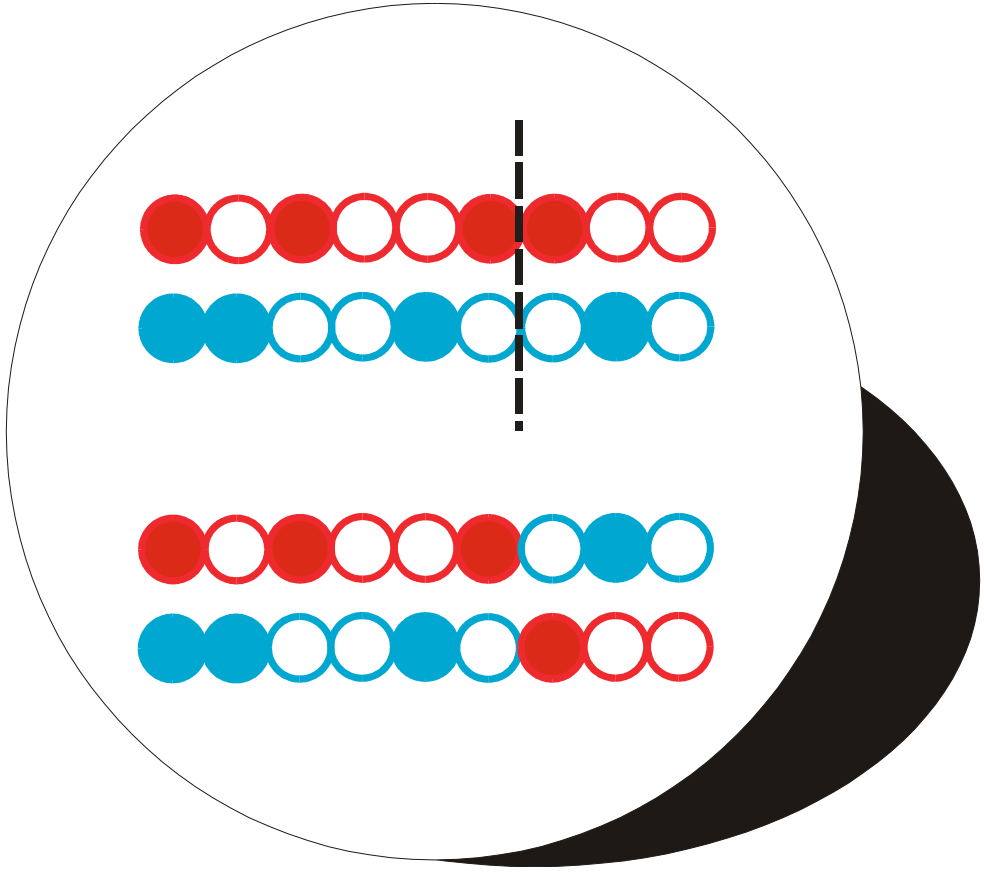
Mate Selection Roulette:

Increasing the likelihood but not guaranteeing the fittest reproduction



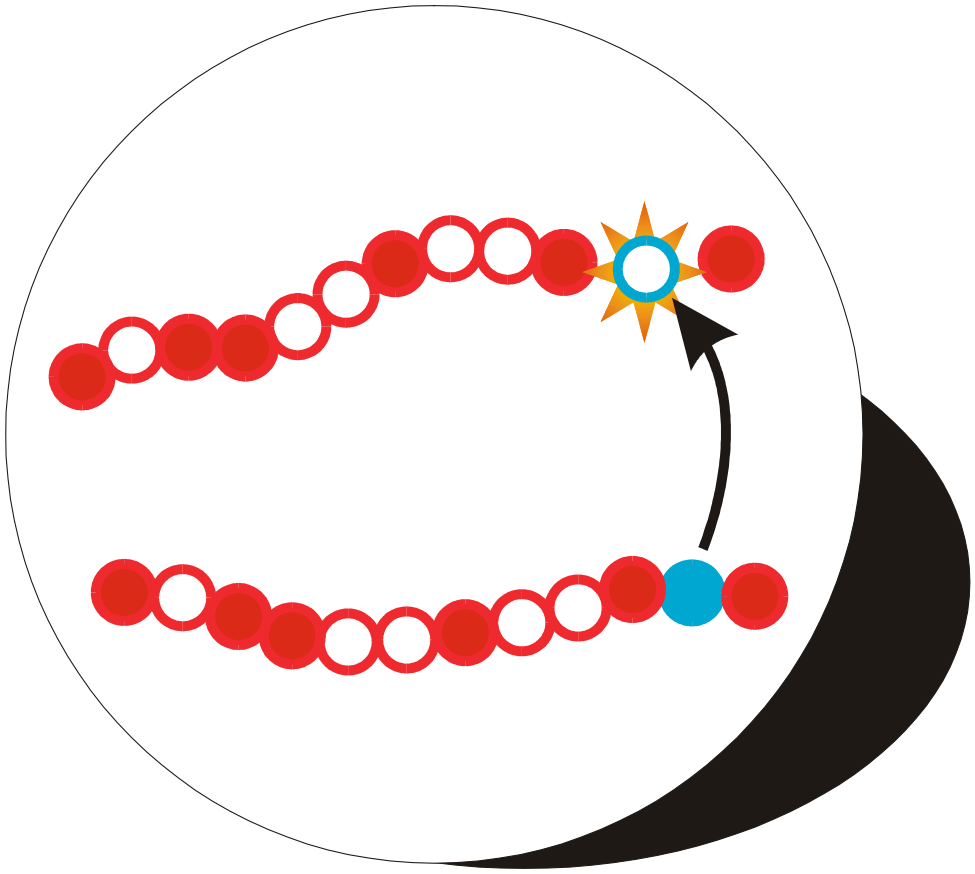
Crossover:

Exchanging information through some part of information (representation)

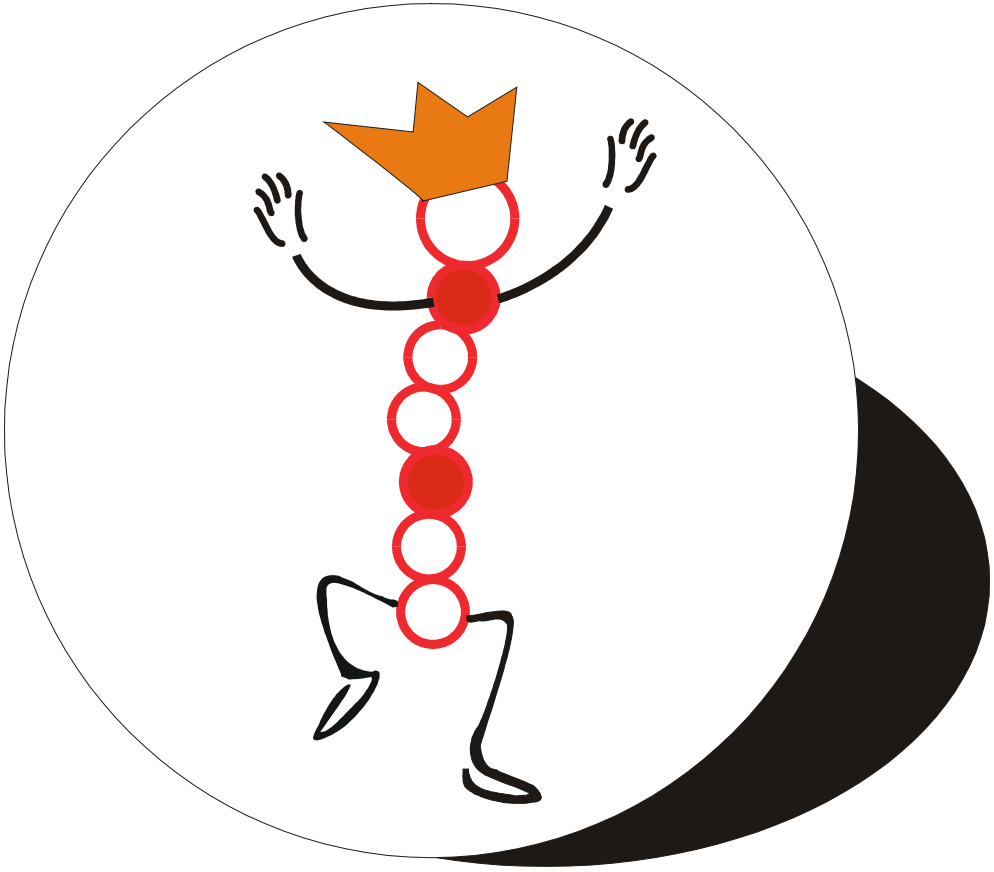


Mutation:

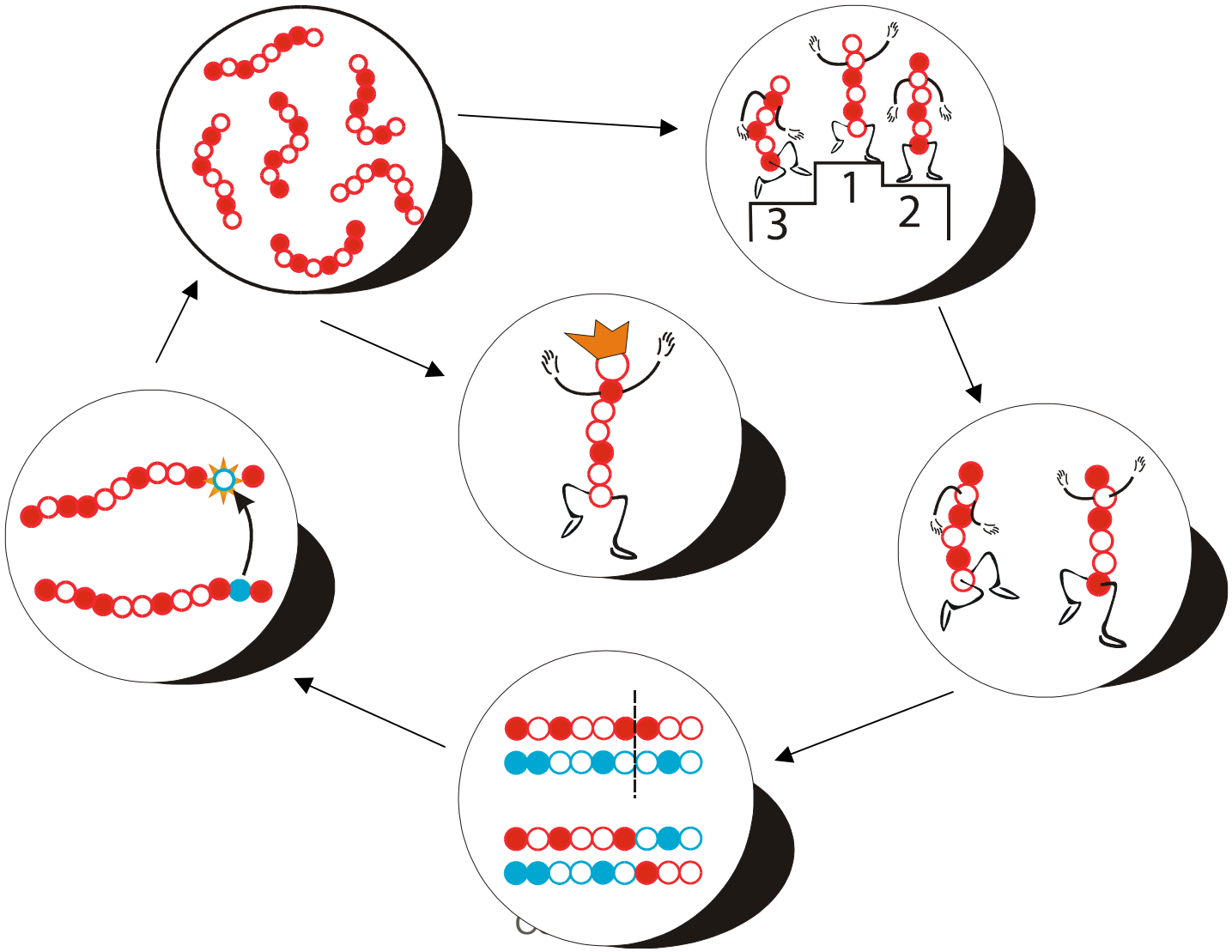
Random change of binary digits from 0 to 1 and vice versa (to avoid local minima)



Best Design



The GA Cycle



Genetic Algorithms

Adv:

•Good to find a region of solution including the optimal solution. But slow in giving the optimal solution

Genetic Approach

- When applied to strings of genes, the approaches are classified as genetic algorithms (GA)
- When applied to pieces of executable programs, the approaches are classified as genetic programming (GP)
- GP operates at a higher level of abstraction than GA

Typical “Chromosome”