Swarm Intelligence: Differential Evolution

- Minimize fitness function $f : R^n \rightarrow R$
- Initialize agents at random positions of the search space
- In each iteration, every agent tries to improve its position:
  - **Pick three other and different agents** (a,b,c)
  - Choose randomly one of the *n* dimensions (i)
  - **Modify dimension i** of the agent, **and also (probably) other** dimensions, by combining the positions of agents a, b, and c

Swarm Intelligence: Differential Evolution

- In each iteration, every agent tries to improve its position:
  - $x_i = a_i + F\,(b_i - c_i)$
  - For each *j* of the rest of dimensions:
    - Generate random number r_j
    - **If rj < CR then $x_j = a_j + F\,(b_j - c_j)$**
  - **If** the fitness of the new **position improves then update** position
- **CR: crossover parameter.** CR in [0,1]
- **F: Differential weight.** F in [0,2]

Swarm Intelligence: Differential Evolution

- Parameter tunning:
  - Population size, number of iterations, CR, F
- Very sensitive!!! See e.g.

https://pdfs.semanticscholar.org/48aa/36e1496c56904f9f6dfc15323e0c45e34a4c.pdf