OPTIMIZATION FOR AI

GLOBAL AND MULTI-OBJECTIVE OPTIMIZATION

Luca Manzoni

TEAM OF THE COURSE

Code:

bpplxhk

The slides and material of the lectures will be uploaded on the Teams of the course

EXAM STRUCTURE

- **Project** + **oral exam**
- Project should be sent about a week before the oral exam
- For December and January there will be some fixed dates for the exam…
- …but after that all exams can be “by appointment”, so take the time that you need

EXAM STRUCTURE

- Kinds of projects:
  - Implement an existing paper and reproduce the results
  - Apply evolutionary algorithms / swarm intelligence algorithms to an existing problem
  - Literature review on a specific topic
- A set of projects will be presented later in the course, but we can discuss personalized project

EXAM STRUCTURE

- Oral exam:
  - First part: presentation (15-20 minutes max) of your project
  - Second part: questions about all topics of the course
- Both parts are essential!

EXAM CHECKLIST

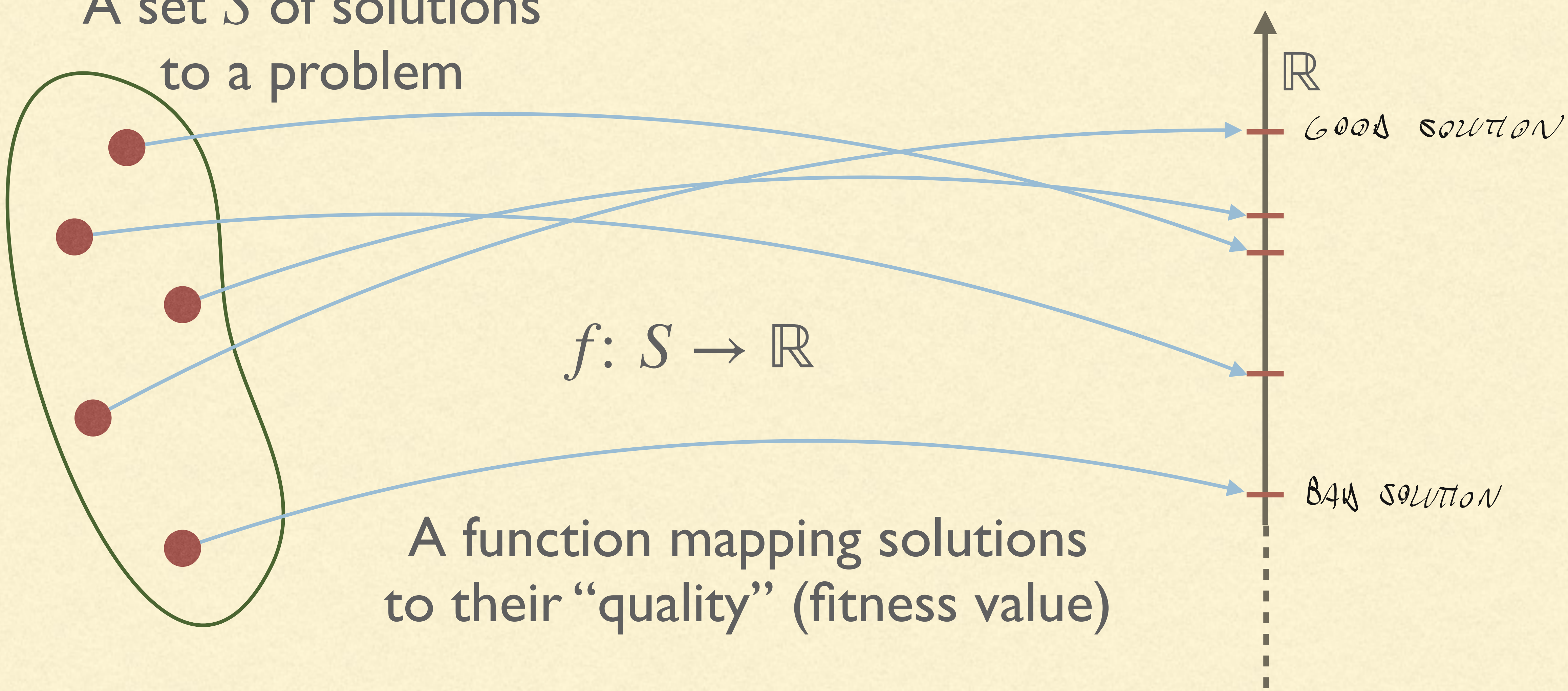
- **Project**
    - Select your project
    - Do your project
    - Fix a date for the oral exam
    - Send the project for evaluation one week before the oral exam
- **Oral Exam**
    - Present your project
    - Answer questions on all the topics of the course

OUTLINE

- What are population-based optimization methods
- Outline of the course
- Reference material
- Genetic Algorithms
- Evolution Strategies

WHAT ARE WE TALKING ABOUT?

A set S of solutions to a problem



A function mapping solutions to their "quality" (fitness value)

We want to find $\operatorname{argmax}_{x \in S} f(x)$ or $\operatorname{argmin}_{x \in S} f(x)$

HOW CAN WE FIND THE OPTIMUM?

TIPICALLY GRADIENT DESCENT IS GOOD ENOUGH

- We might be unable to solve the problem analytically
- S might be too large to perform an exhaustive search
- We might have very few assumption on f, i.e., f is a "black box"
- It might be OK to return "good enough" solutions

A TRIVIAL PROBLEM: ONEMAX

- Let $S = \{0,1\}^n$, hence the size of the space is 2^n
- Let $f(x) = $ # of ones in x
- We want to maximise f
- Clearly, the optimum is 1^n with fitness value n

EX. WITH $n=5 \Rightarrow 01001$
$f(01001) = 2$
THE $\max_x f(x)$ IS $x = 11111$

APPROACH #1: RANDOM SEARCH

- Select a solution b from S (it is not important how)
- Repeat the following until some termination criteria is met
  - Let x be a solution selected uniformly at random from S
  - If $f(x) \geq f(b)$ substitute b with x
- Return b

APPROACH #1: RANDOM SEARCH

- Even if we avoid sampling the same solution more than once, we will still need to explore a significant fraction of the search space
- Without repetitions, it is like exhaustive search for some choice of the order of enumeration of the solutions
- Generally unfeasible

RANDOM START THE INIZIALIZATION
OR TO APPLY A CHAOS FACTOR
TO THE RESEARCH

- In **some** search spaces this is better than other kinds of search…
- …but hopefully this is not something that happens for real problems

APPROACH #2: HILL CLIMBING

- Let b be an initial solution from S
- Repeat the following until some termination criteria is met
  - Let x be a **neighbor** of b

    ↳ NOT A RANDOM ELEMENT FROM S
  - If $f(x) \geq f(b)$ then replace b with x
- Return b

A GRADIENT DESCENT
WITHOUT COMPUTING
THE EIGHENVALUE

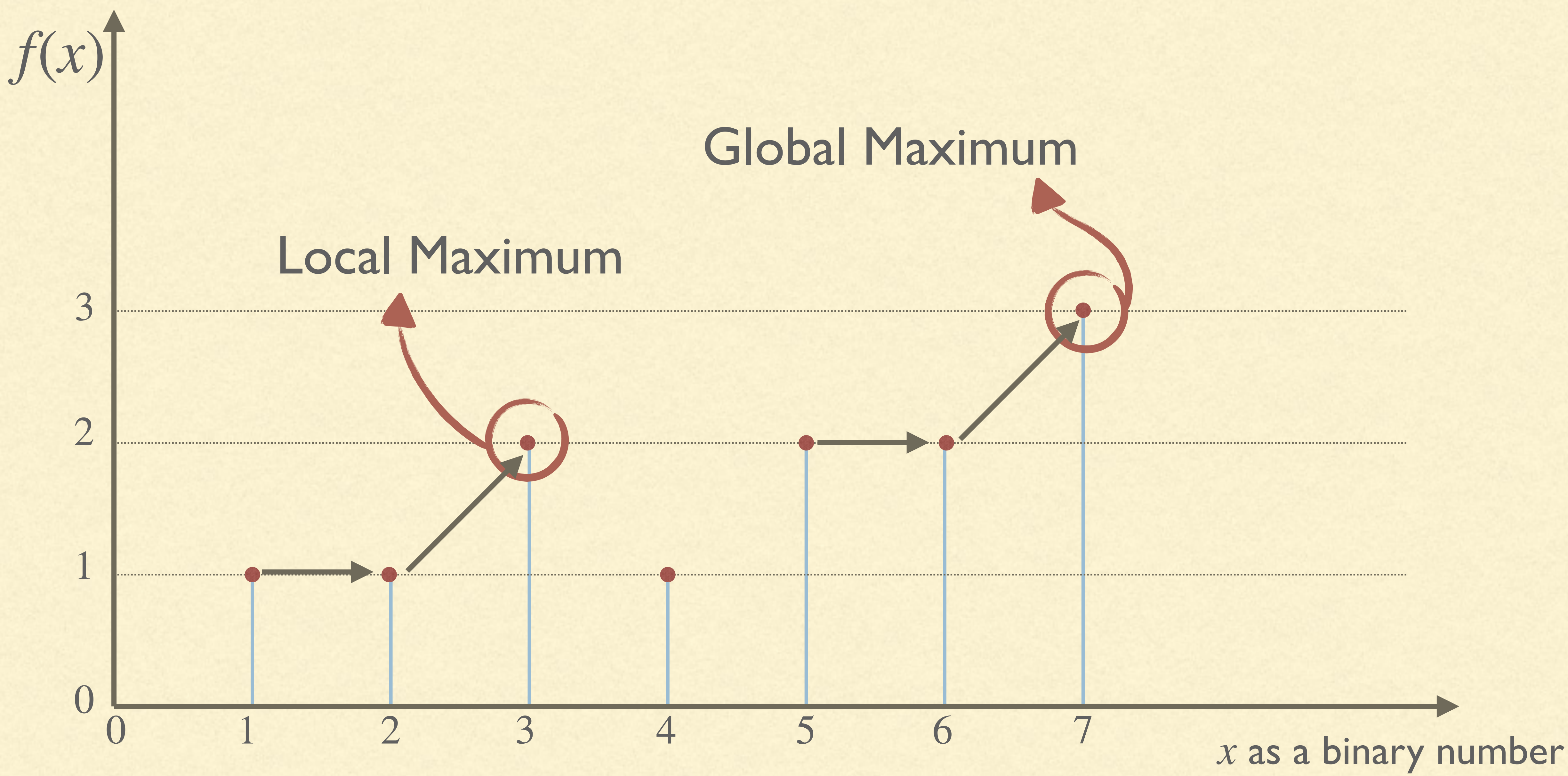
APPROACH #2: HILL CLIMBING

- We are defining a neighborhood structure on S

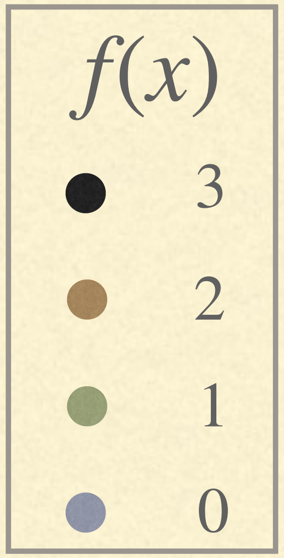
AND HOW WE DEFINE THE PROBLEM

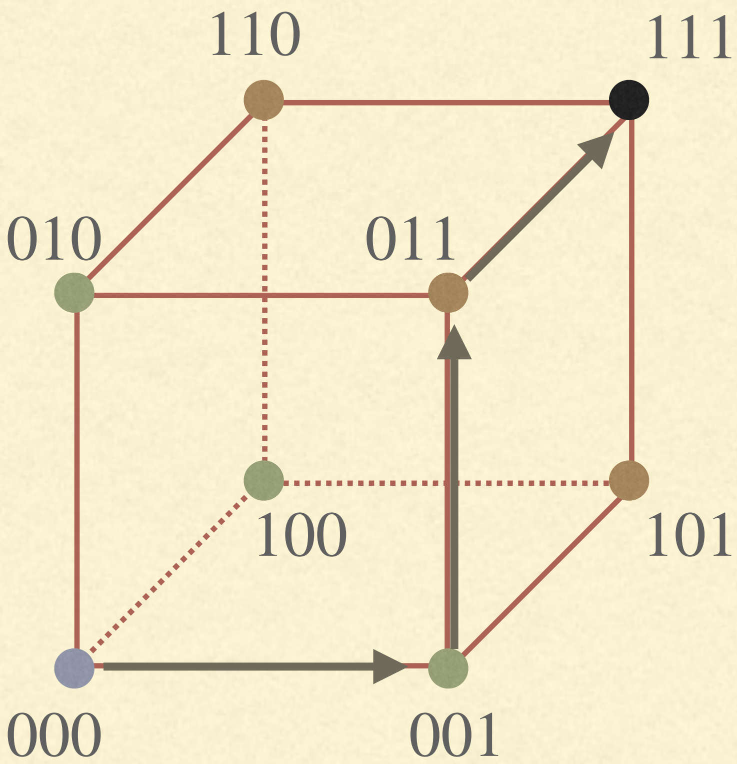
- The ability to find a global optimum depends on this structure:
    - OneMax with neighborhood of x given by $x + 1$ and $x - 1$
    - OneMax with neighborhood of x given by all binary strings at Hamming distance one.

LOCAL AND GLOBAL MAXIMA



LOCAL AND GLOBAL MAXIMA





Notice that there are no local optima

APPROACH #3: SIMULATED ANNEALING

THE CONCEPT OF SIMULATED ANNEALING IS "BEATING THE IRON WHILE IS HOT"

- Let b be an initial solution from S and T the "temperature"
- Repeat the following until some termination criteria is met
  - Let x be a **neighbor** of b
  - If $f(x) \geq f(b)$ then replace b with x
  - Otherwise replace b with x with probability $e^{\frac{f(x)-f(b)}{T}}$
  - Update T (by decreasing it) according to some schedule
- Return b

IT IS AN ANALOGY OF PYSHICAL SYSTEM. THE HIGHER THE TEMPERATURE (MORE ENERGY) THE HIGHER THE PROBABILITY.

APPROACH #3: SIMULATED ANNEALING

- The idea is to allow the selection of less fit solutions with some probability that depends from the time and the difference in fitness
- This allows to reduce the risk of getting stuck in a local optimum
  BUT ALLOWS TO MOVE FREELY AROUND THE SOLUTION SPACE UNTIL IS "HOT"
- The choice of the schedule is important!

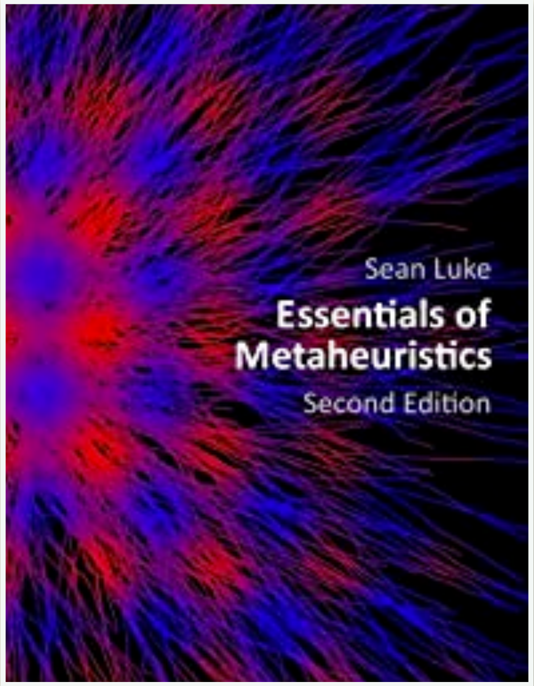
MULTIPLE RESTARTS

- For Hill Climbing and Simulated Annealing we can reduce the risk of getting stuck on a local optimum by repeating the process several times
- Each repetition is independent…
- …can we do better?
- The idea is to use a multiset of solutions instead of working with one solution at a time
- This time the different solutions "interact" in some way

OUTLINE OF THE COURSE

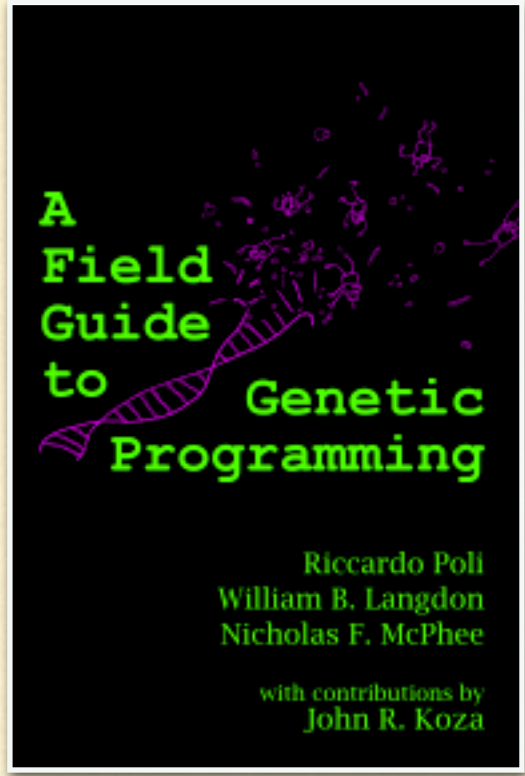
- Genetic Algorithms
- Evolution Strategies
- Genetic Programming
- Particle Swarm Optimization and Ant-Colony Optimization
- Differential Evolution
- Neuroevolution => BASED ON NEURAL NETWORKS
- EDA and CMA-ES
- Parallel implementations
- Multi-objective optimization
- Coevolution
- Policy Optimization
- Theory of Evolutionary Computation

REFERENCE MATERIAL

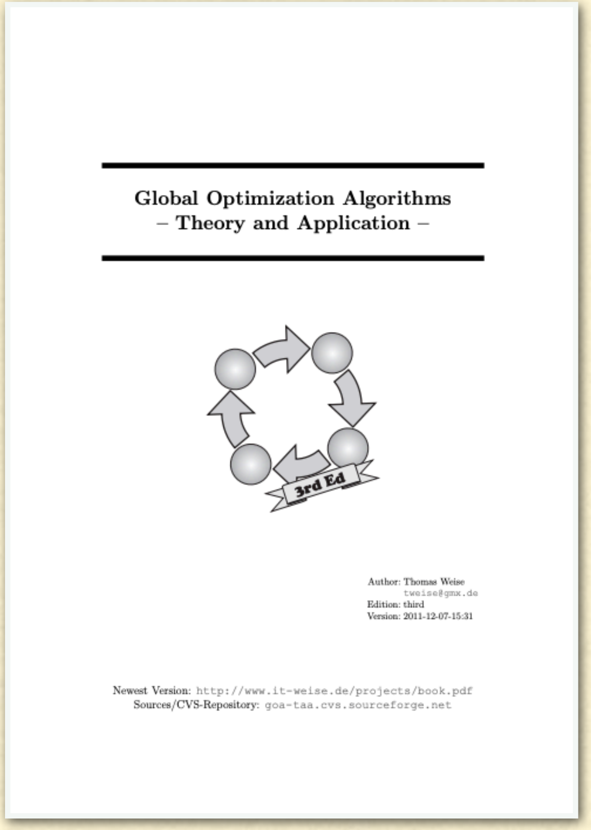


S. Luke
Essentials of Metaheuristics, 2nd Edition
https://cs.gmu.edu/~sean/book/metaheuristics/

R. Poli, W. R. Langdon, N. F. McPhee
A Field Guide to Genetic Programming
http://www.gp-field-guide.org.uk



REFERENCE MATERIAL



T. Weise
Global Optimization Algorithms – Theory and Application, 3rd Edition
(Search for it online)

F. Neumann and C. Witt
Bioinspired Computation in Combinatorial Optimization
Algorithms and Their Computational Complexity
http://www.bioinspiredcomputation.com/self-archived-bookNeumannWitt.pdf