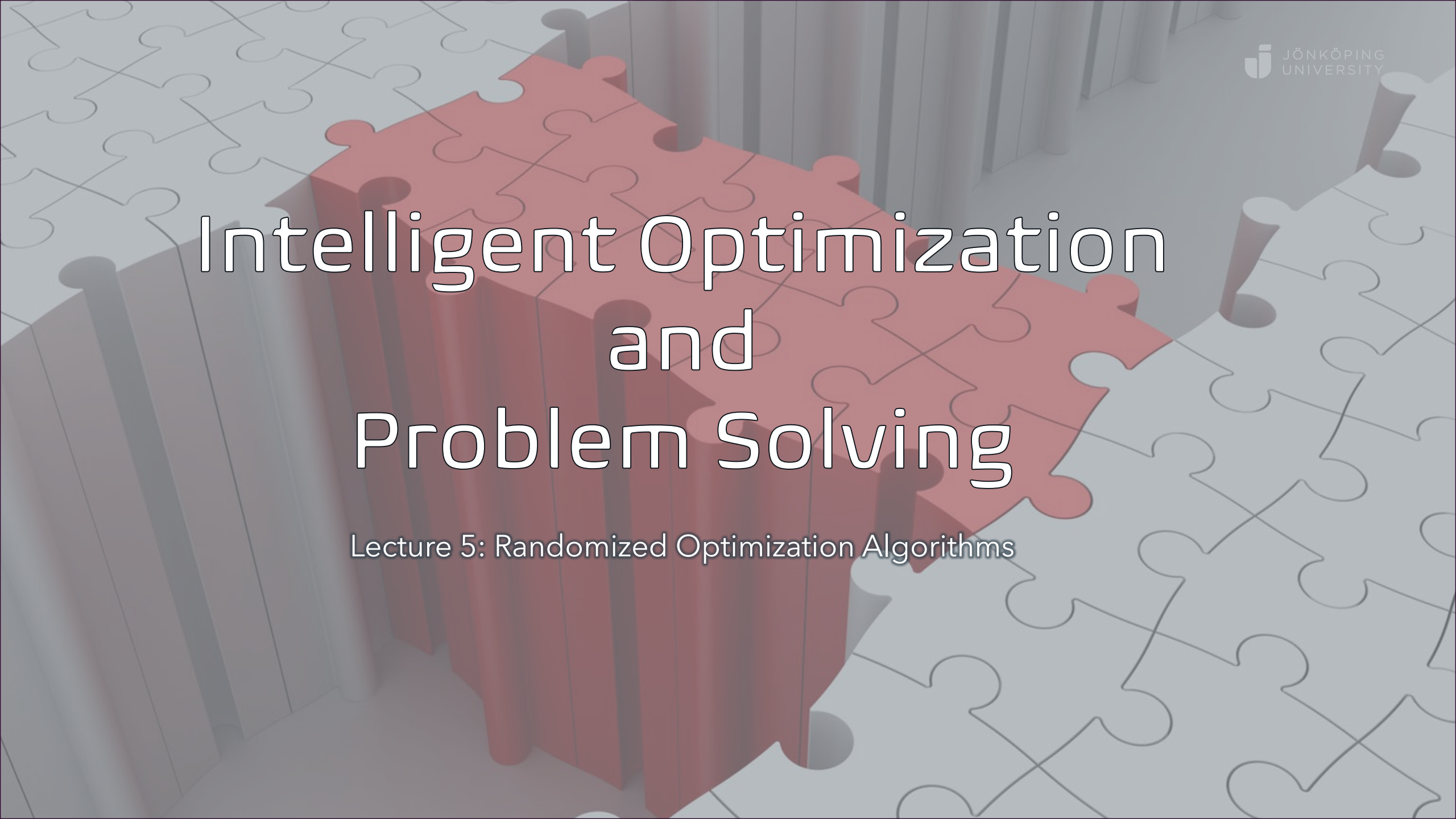


JÖNKÖPING UNIVERSITY

Intelligent Optimization and Problem Solving

Lecture 5: Randomized Optimization Algorithms

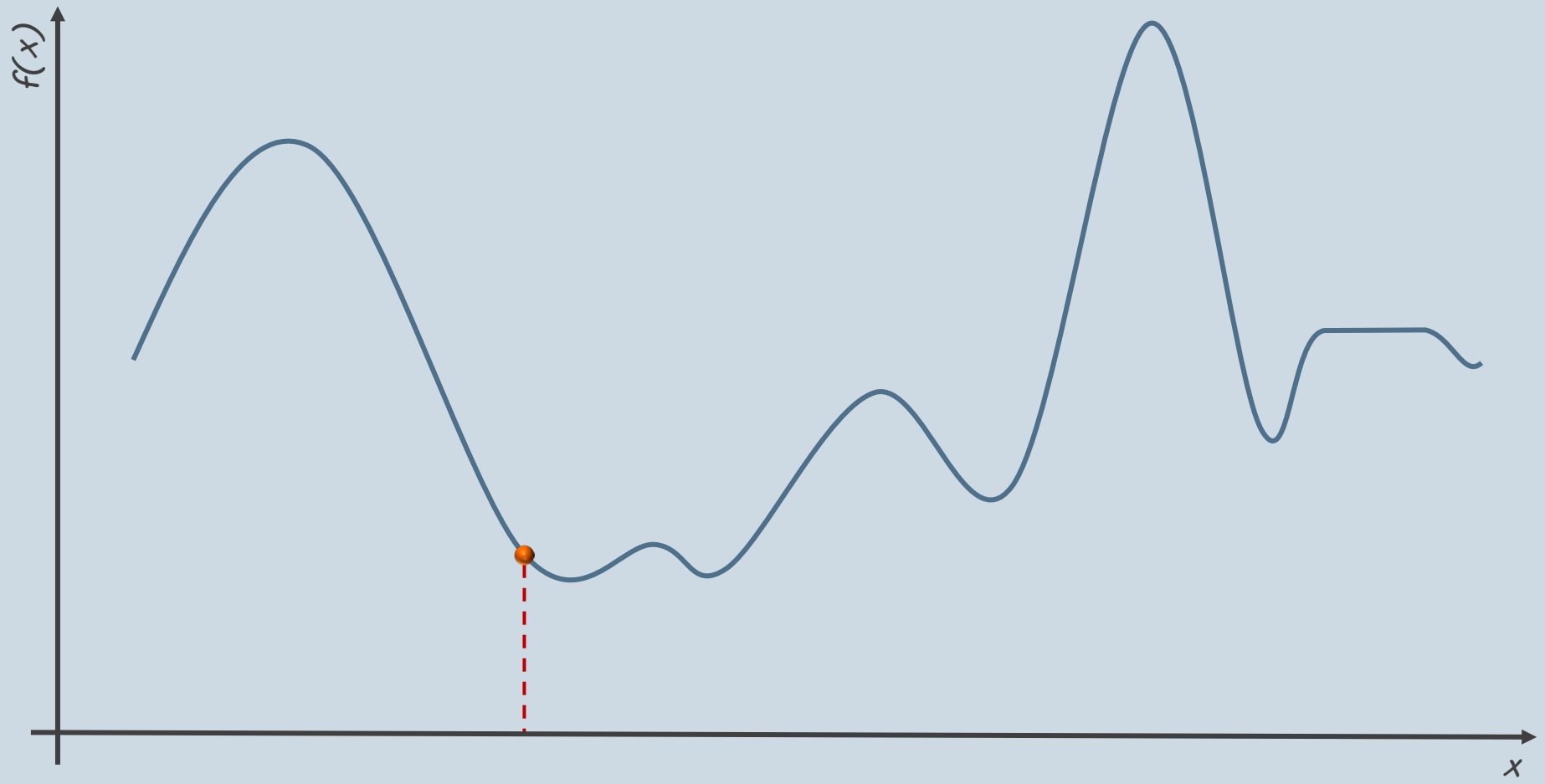
JÖNKÖPING UNIVERSITY

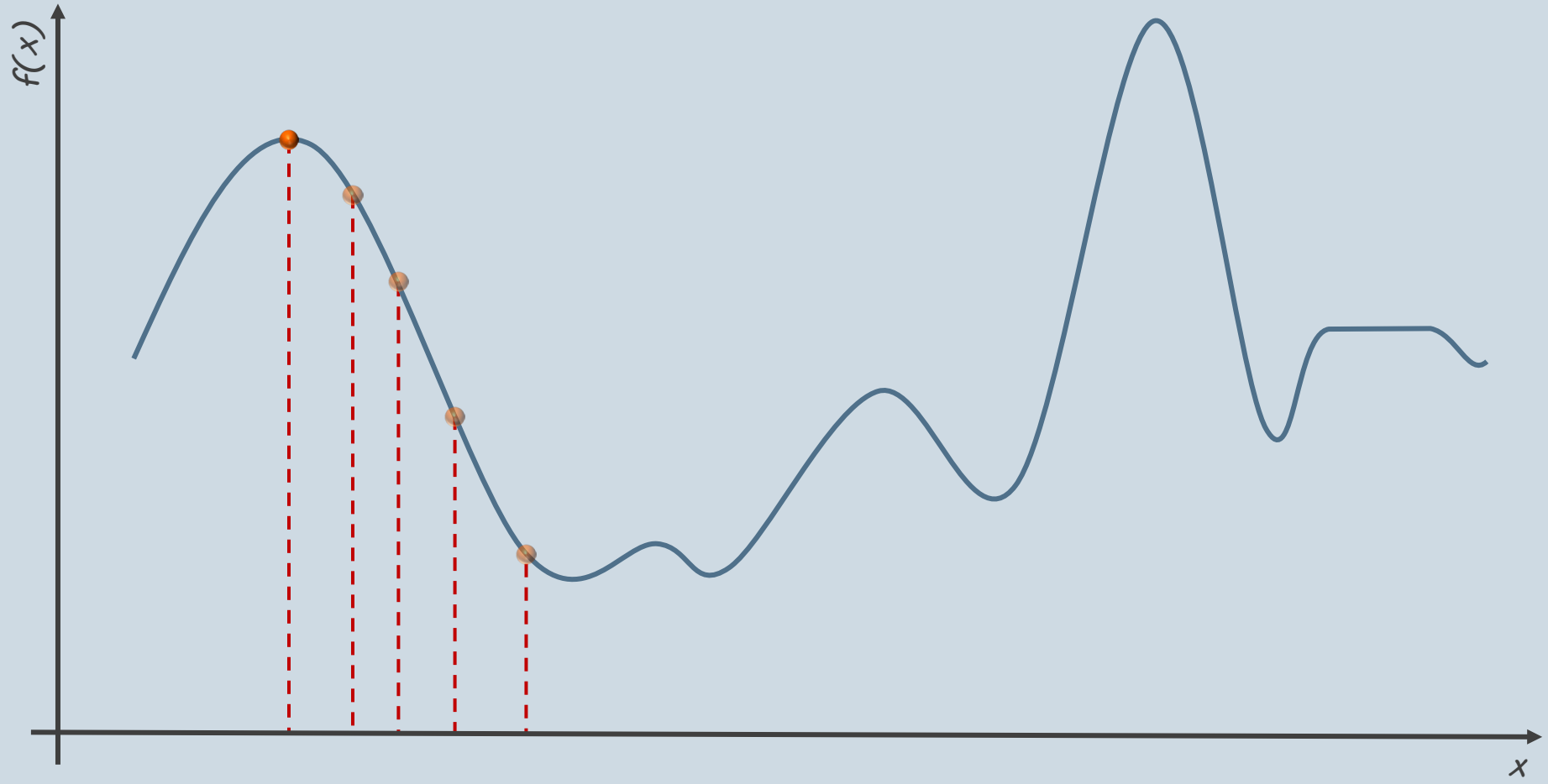
Problem solving

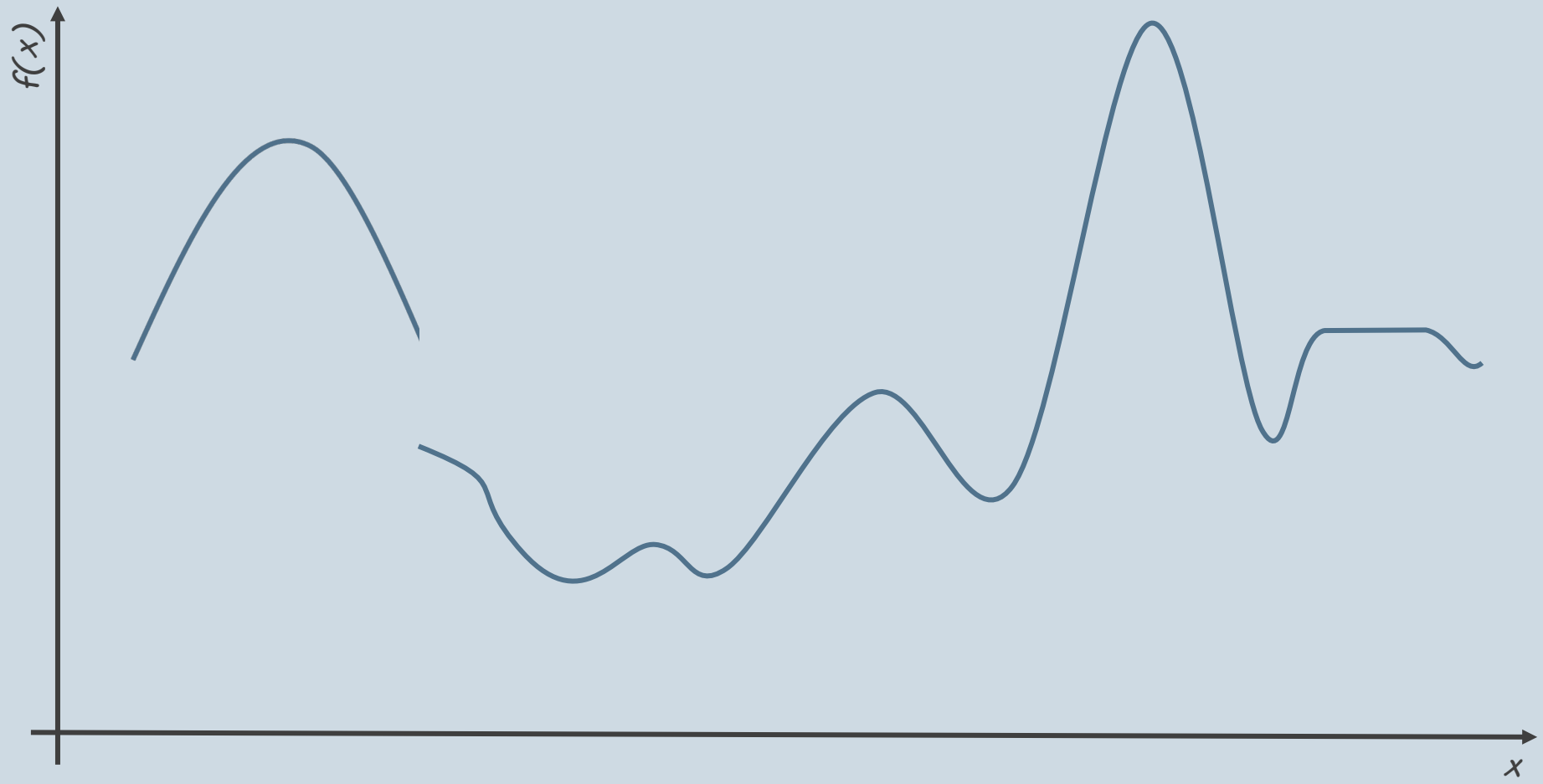
How do we define a problem?

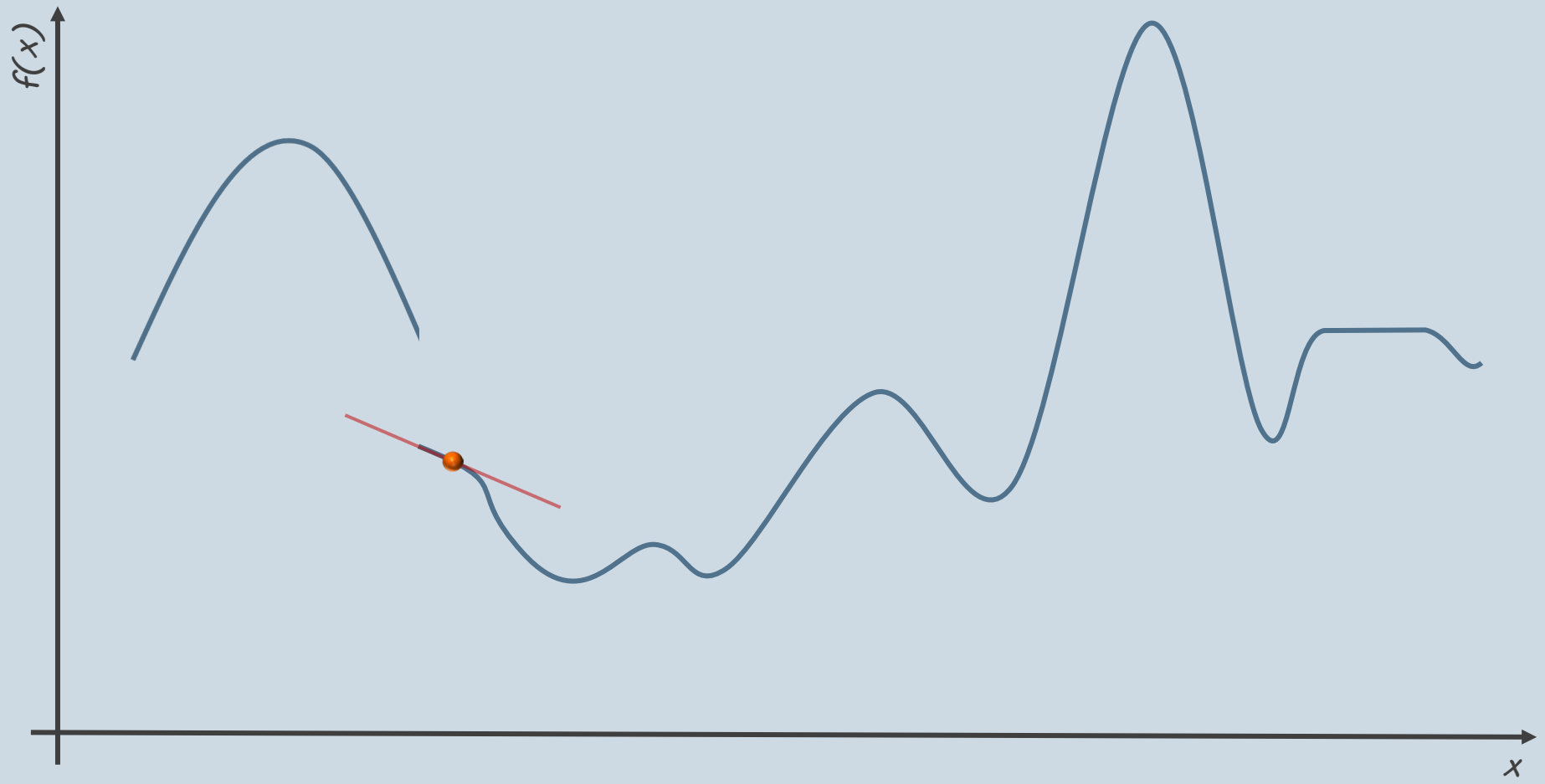
- objective
- constraints

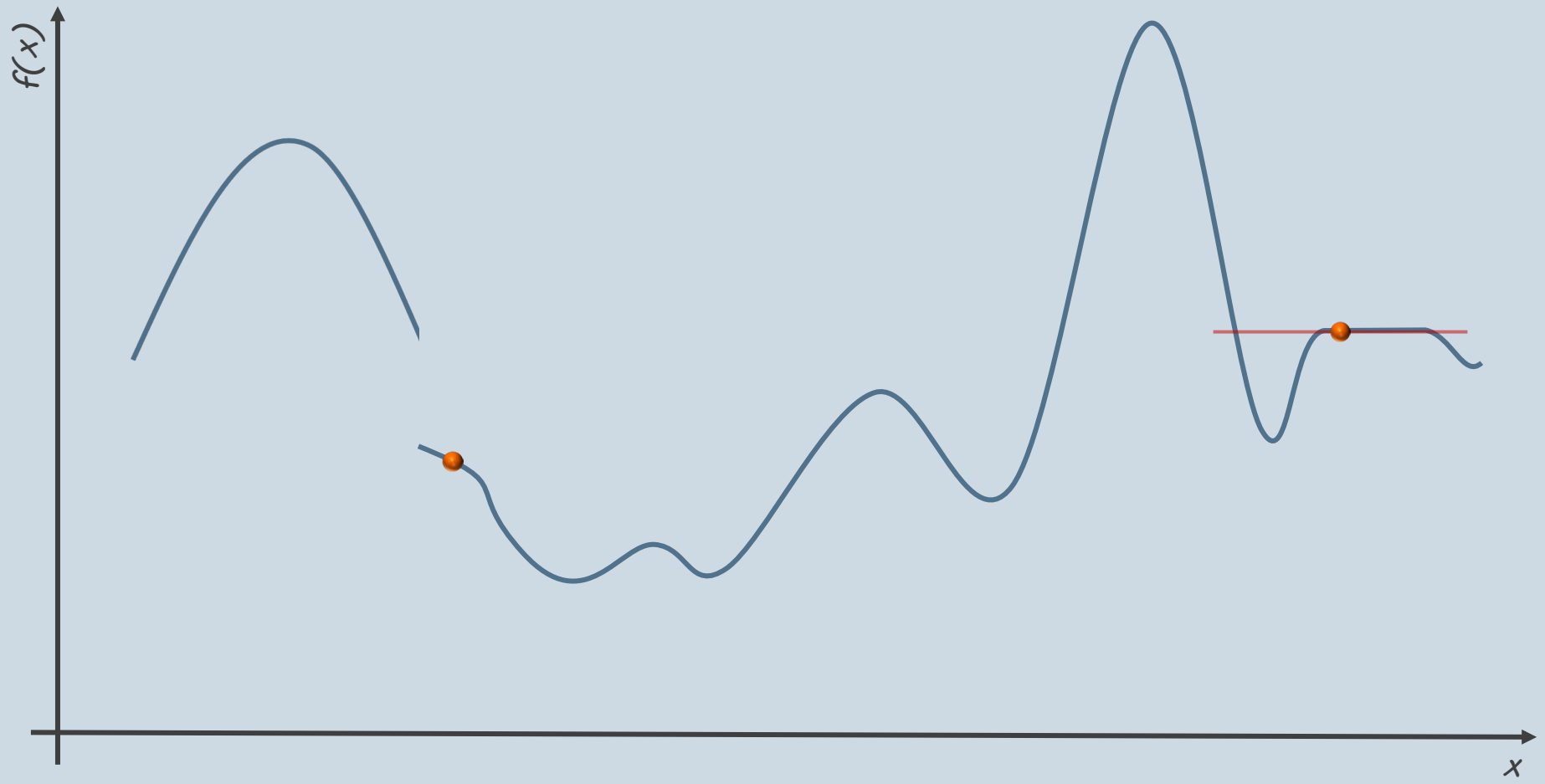
How do we solve a problem?

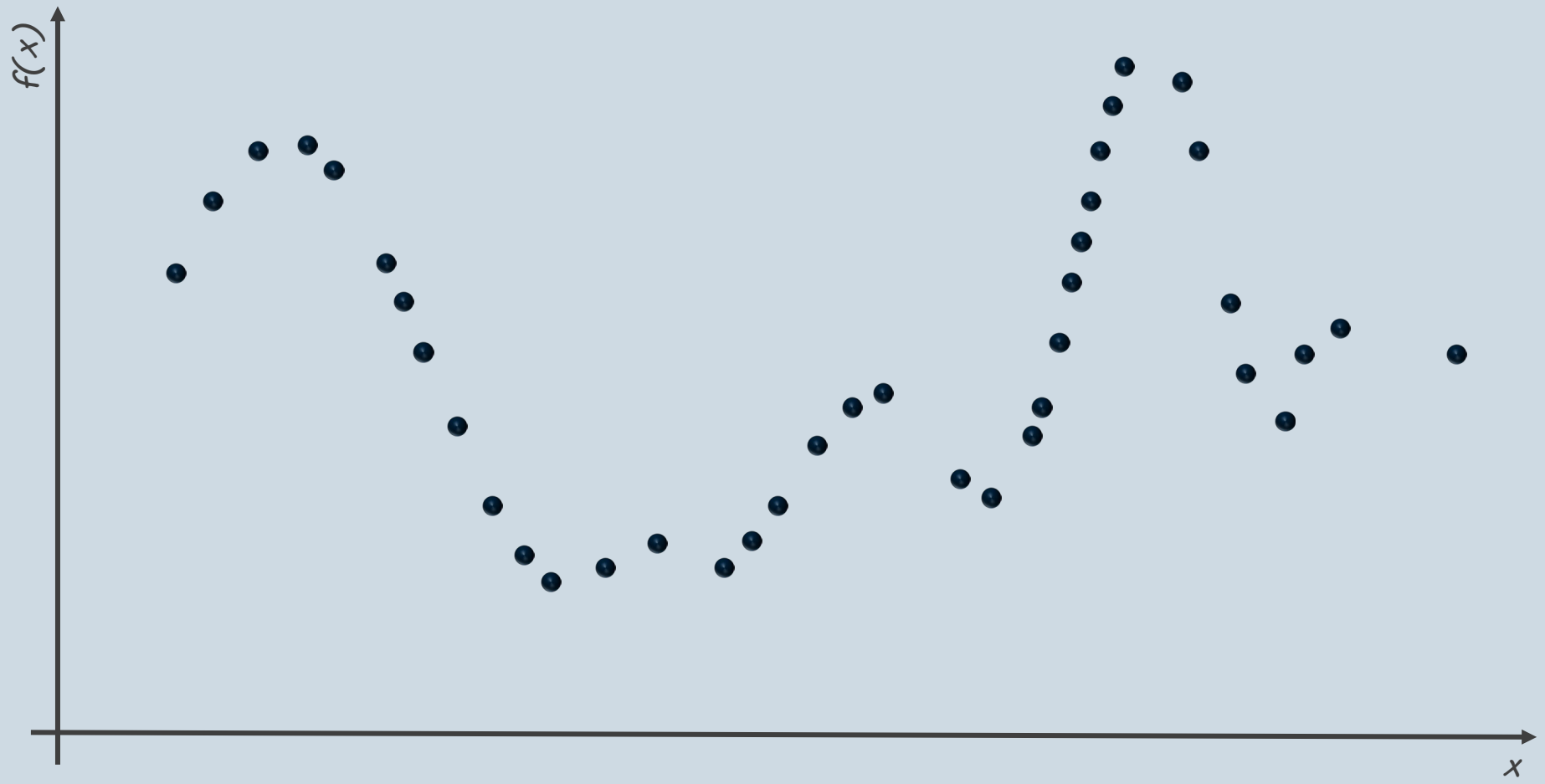

f(x)
x


f(x)
x

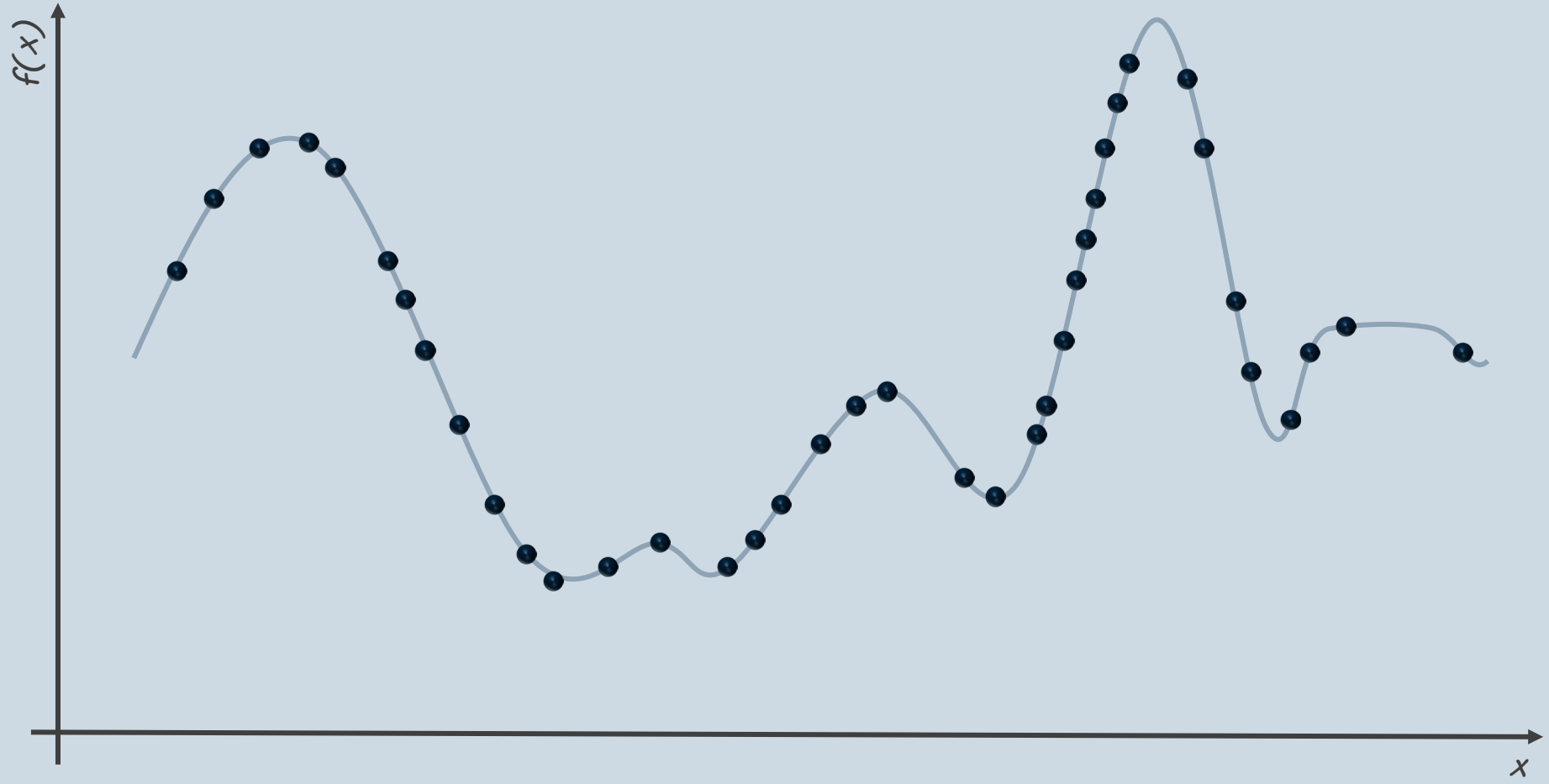

f(x)
x


f(x)
x


f(x)
x


f(x)
x

How much should we sample?



JÖNKÖPING UNIVERSITY

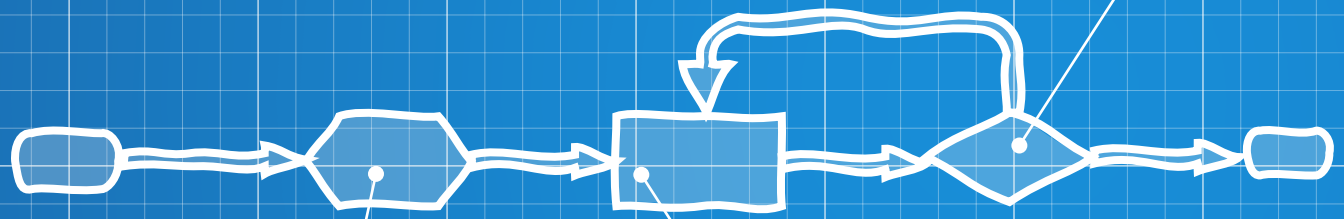
Randomized Optimization Algorithms

Randomized optimization algorithms (ROAs) are techniques of stochastic rather than deterministic state transitions, even when the optimization problem itself is deterministic.

Heuristics are techniques designed to find an approximate solution when traditional methods fail to find an exact one.

Meta-heuristics are high-level, problem-independent frameworks that guide the search for solutions in optimization problems with large solution spaces.

Meta-heuristics



Stopping criterion

- number of steps (iterations)
- threshold

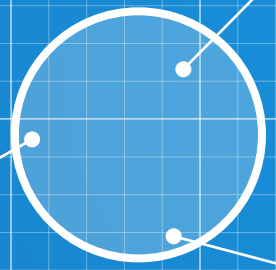
Initialization

a) candidate solution(s)
b) parameters

Searching phase

the algorithm's search process

Candidate solution
(individual)



Format
- vector
- arbitrary-length list of objects
- unordered set of objects
- graph

Data structure
- position
- memory
- functions
- measures/metrics

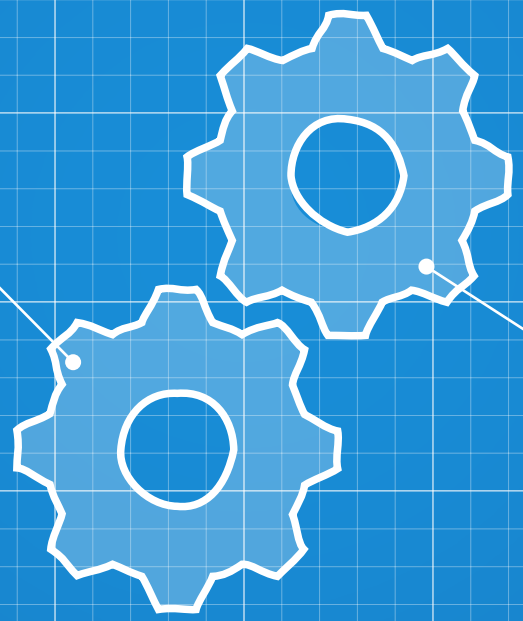
Representation
- ✓ completeness
- ✓ (computational) efficiency
- ✓ connexity

JÖNKÖPING UNIVERSITY

Search mechanisms (operators)

Mechanisms consideration

- *one*
- *multiple*
- *hybrids*



New candidate solutions

How is a new solution created?

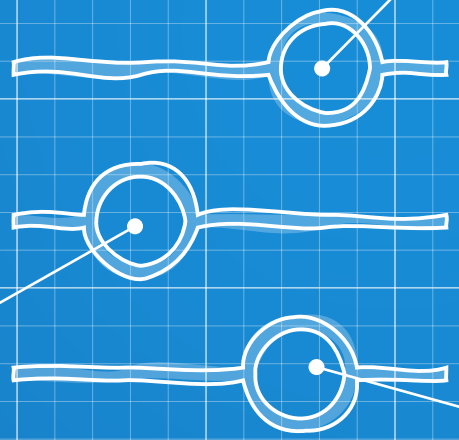
- *randomly*
- *based on another one*
- *influenced by several other ones*

JÖNKÖPING UNIVERSITY

Parameters

Step size

how far will the new candidate solution be?



Effect on

- performance
- behavior

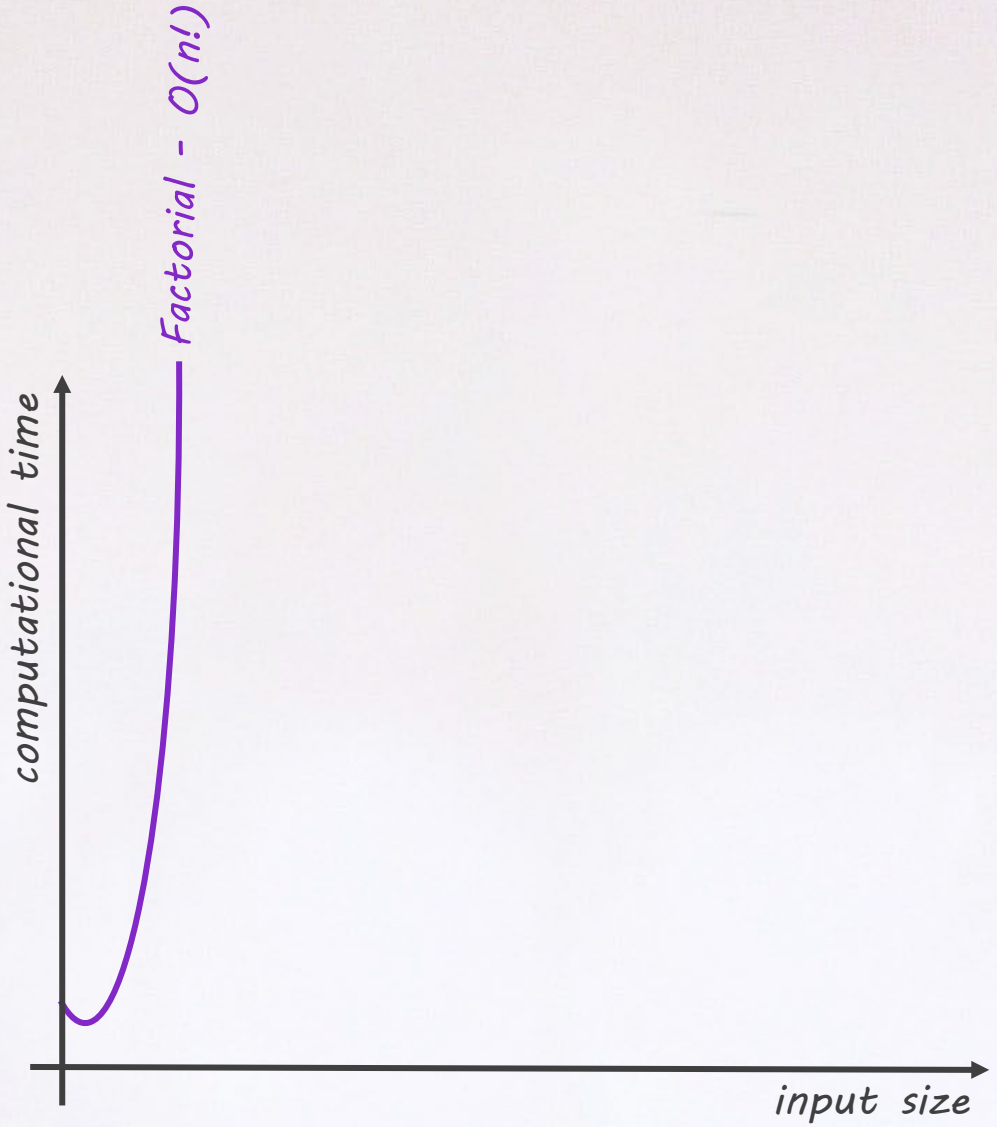
Stopping criteria

when searching doesn't lead to a better solution?

Complexity

e.g., Traveling Salesman Problem (TSP)

$O(n!)$



n: number of cities

why for small input, we don't see these trends?

Dr. Alexandros Tzanetos

JÖNKÖPING UNIVERSITY
School of Engineering

Complexity

e.g., Traveling Salesman Problem (TSP)

$O(n!)$

n: number of cities

why for small input, we don't see these trends?

Dr. Alexandros Tzanetos

JÖNKÖPING UNIVERSITY
School of Engineering

Complexity

e.g., Traveling Salesman Problem (TSP)

$O(n!)$

some randomized algorithm A
- a P number of candidate solutions
- of n length
- for T number of iterations

$O(P \cdot n \cdot T)$

n: number of cities

why for small input, we don't see these trends?

Dr. Alexandros Tzanetos

JÖNKÖPING UNIVERSITY
School of Engineering

Complexity

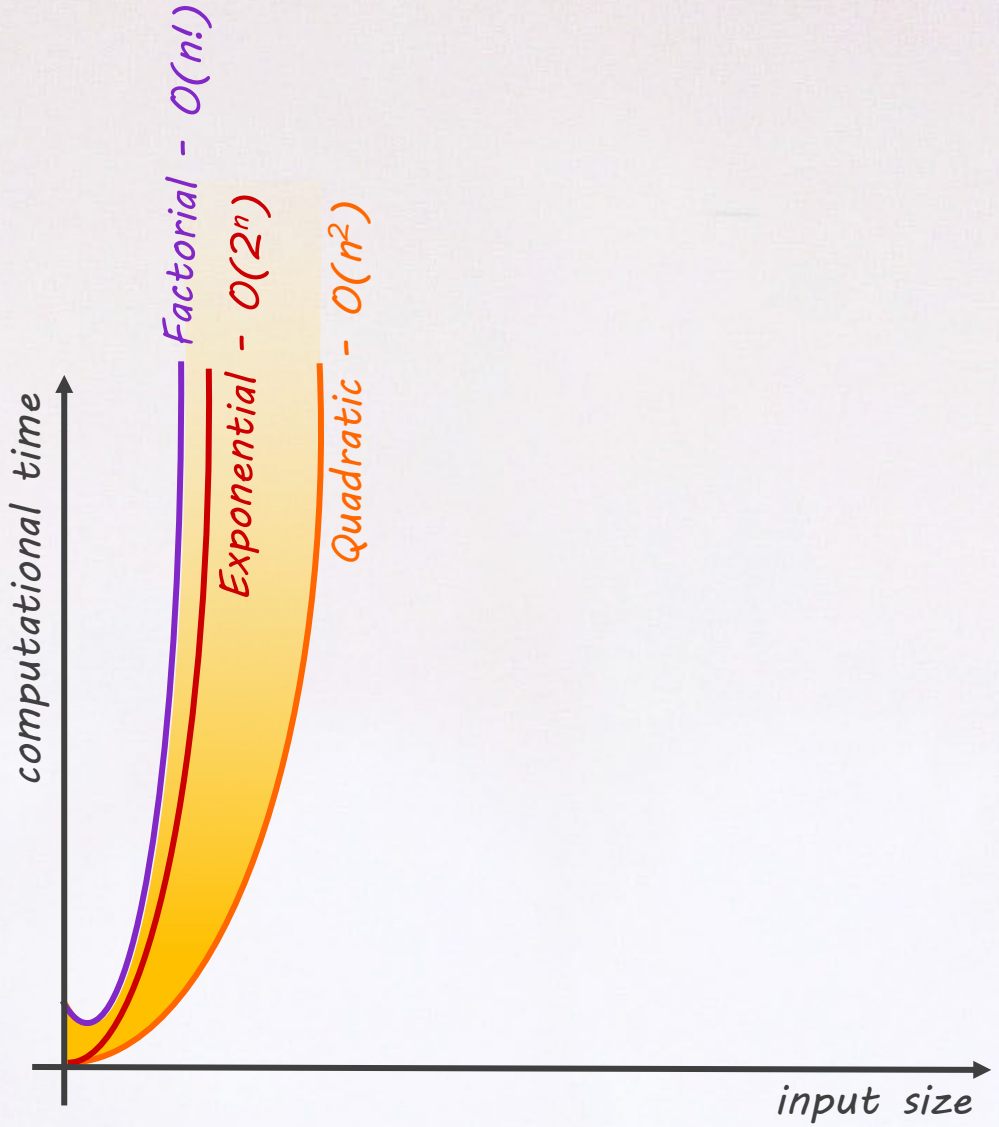
e.g., Traveling Salesman Problem (TSP)

$O(n!)$

some randomized algorithm A
- a P number of candidate solutions
- of n length
- for T number of iterations

$O(P \cdot n \cdot T)$

n: number of cities



why for small input, we don't see these trends?

Dr. Alexandros Tzanetos

JÖNKÖPING UNIVERSITY
School of Engineering

Complexity

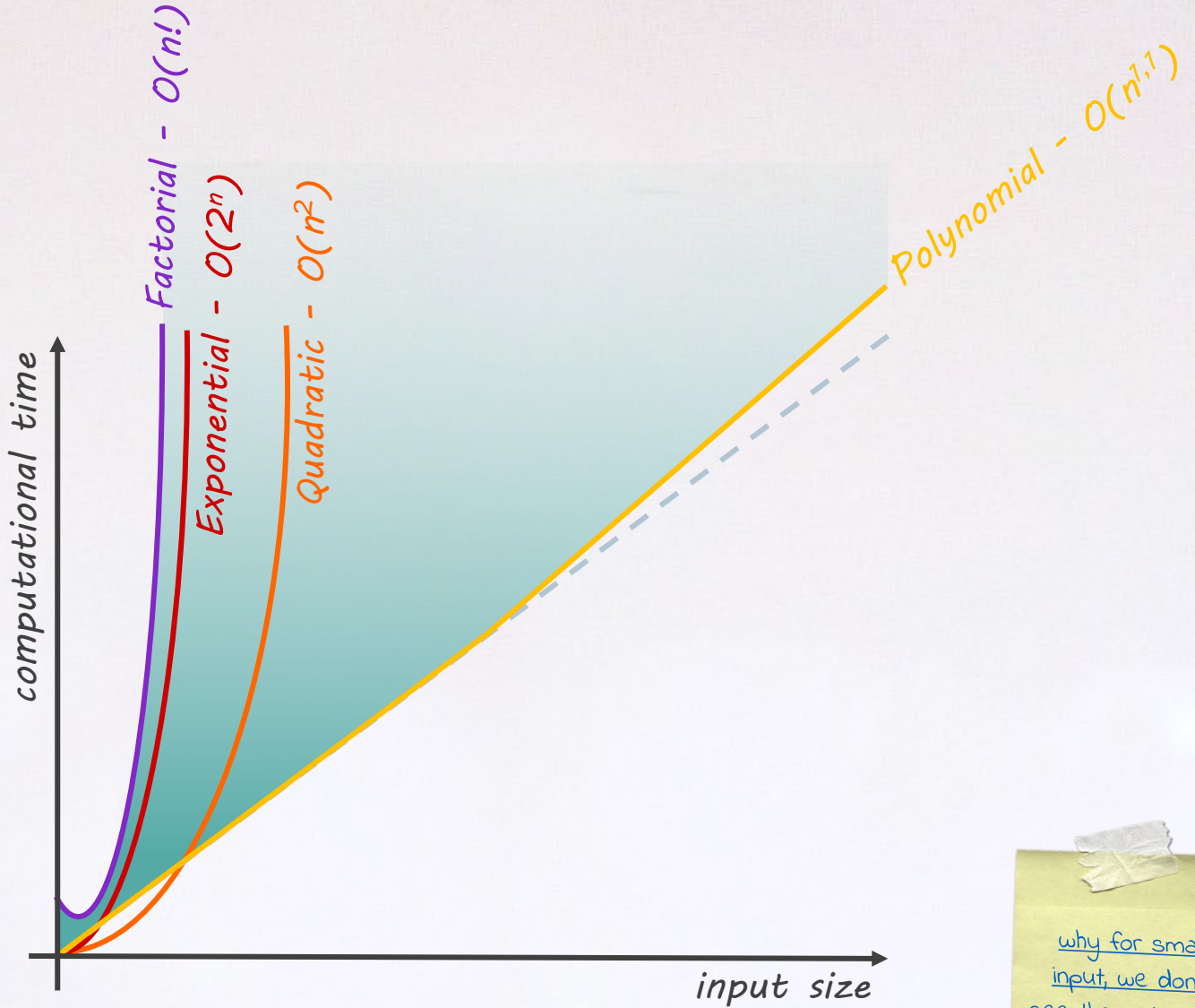
e.g., Traveling Salesman Problem (TSP)

$O(n!)$

some randomized algorithm A
- a P number of candidate solutions
- of n length
- for T number of iterations

$O(P \cdot n \cdot T)$

n: number of cities



why for small input, we don't see these trends?

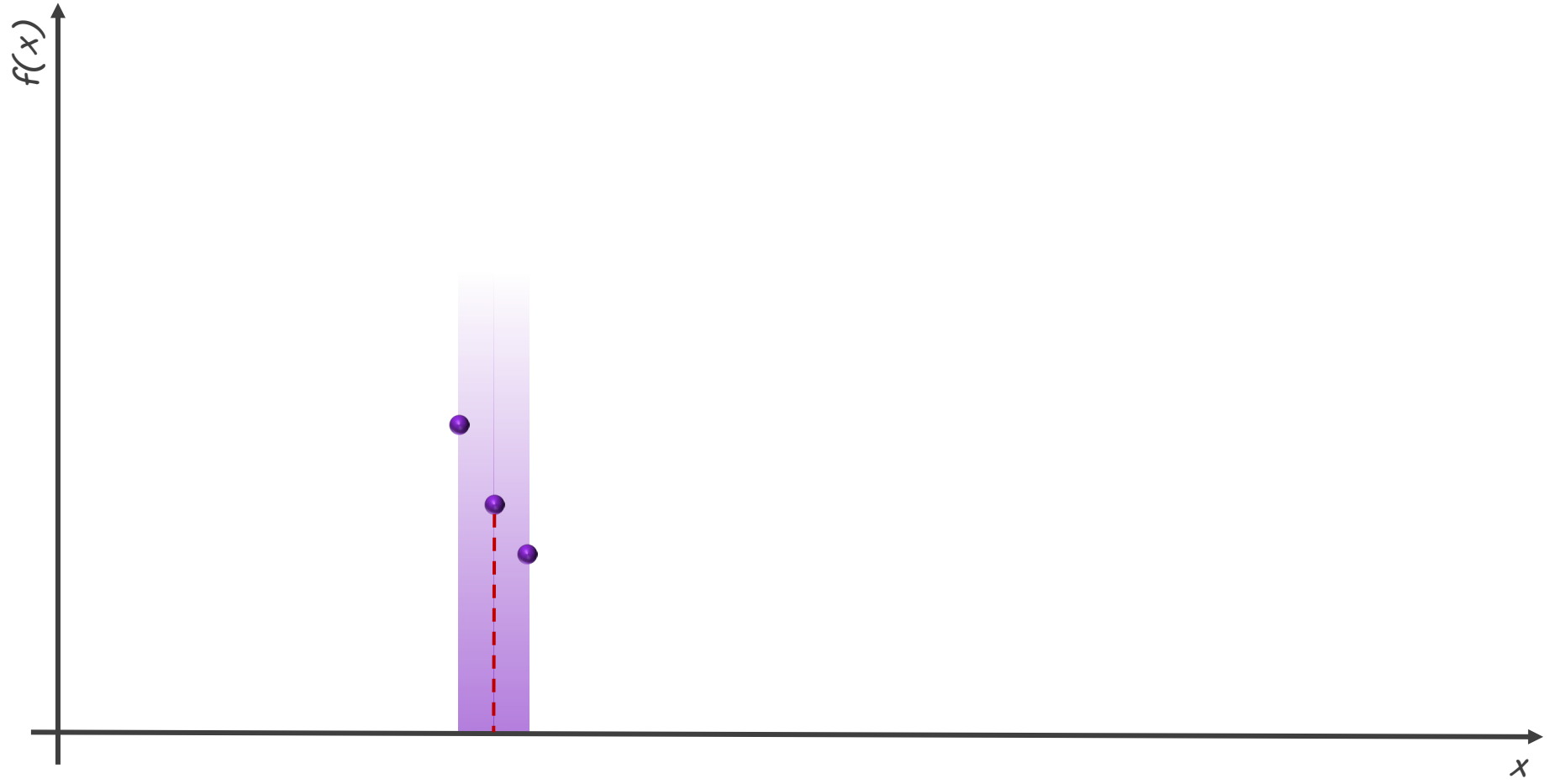
Dr. Alexandros Tzanetos

JÖNKÖPING UNIVERSITY
School of Engineering

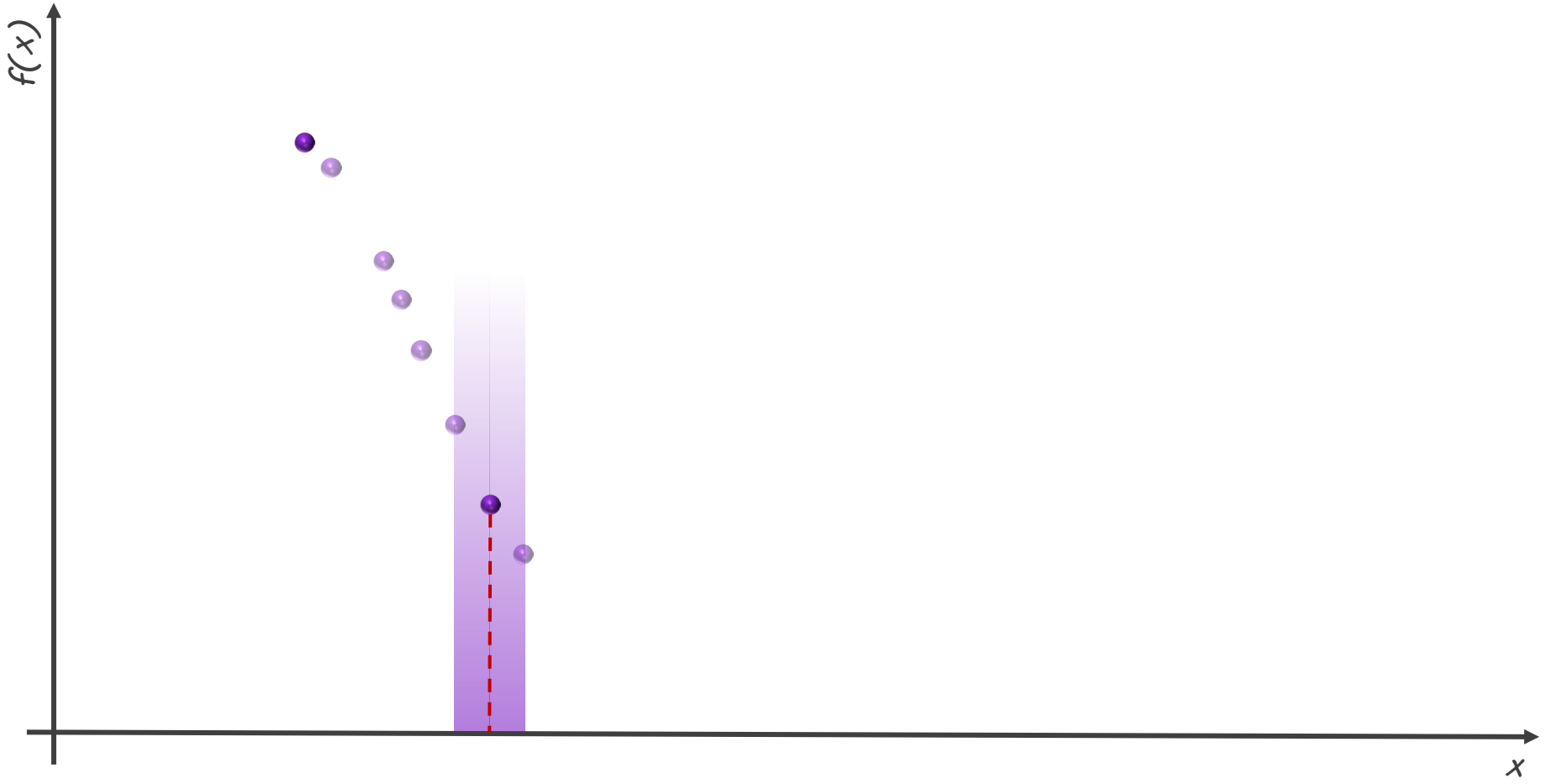
Stochastic optimization

Dr. Alexandros Tzanetos

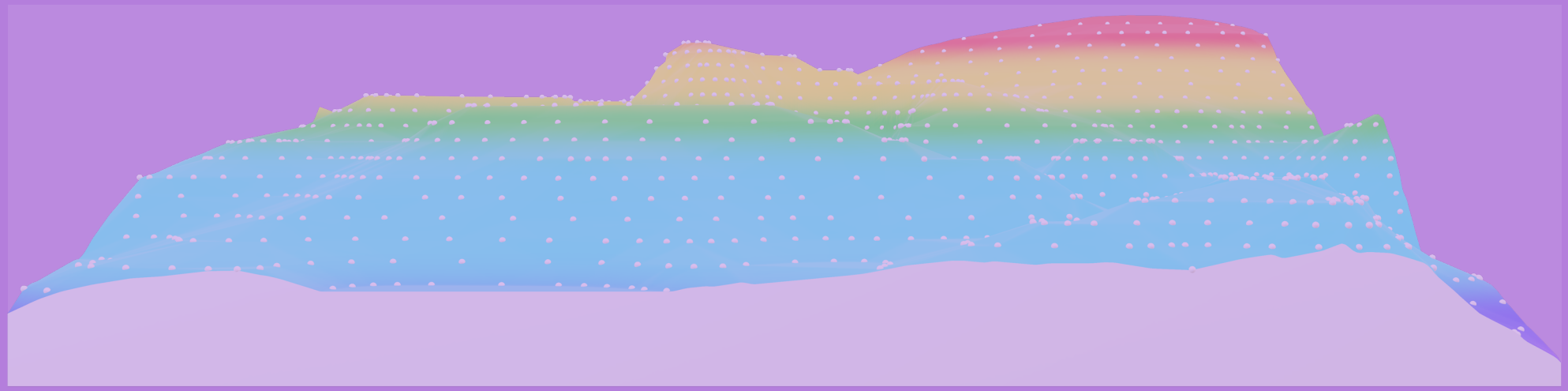
Steepest Ascent Hill-Climbing

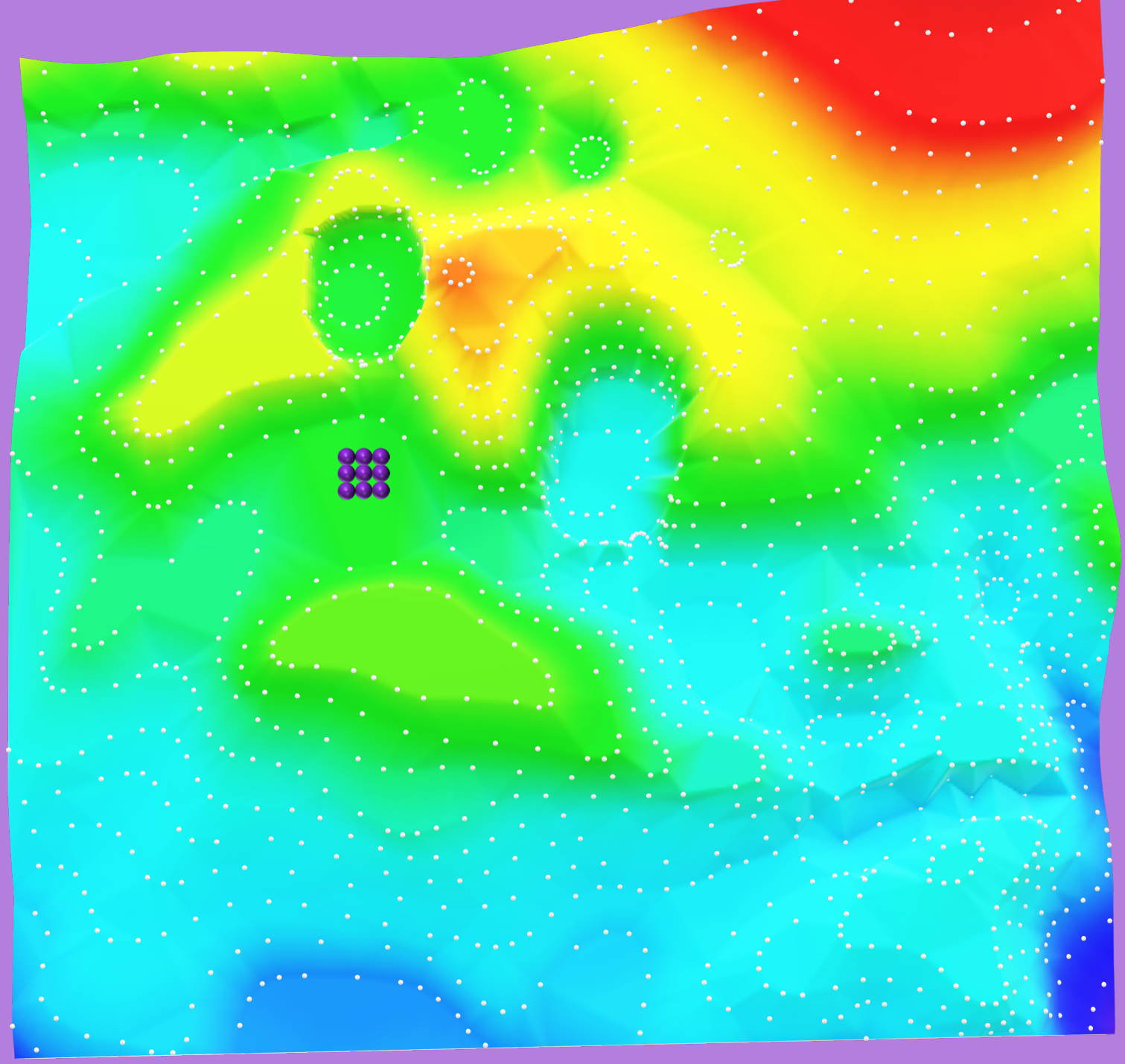


Steepest Ascent Hill-Climbing



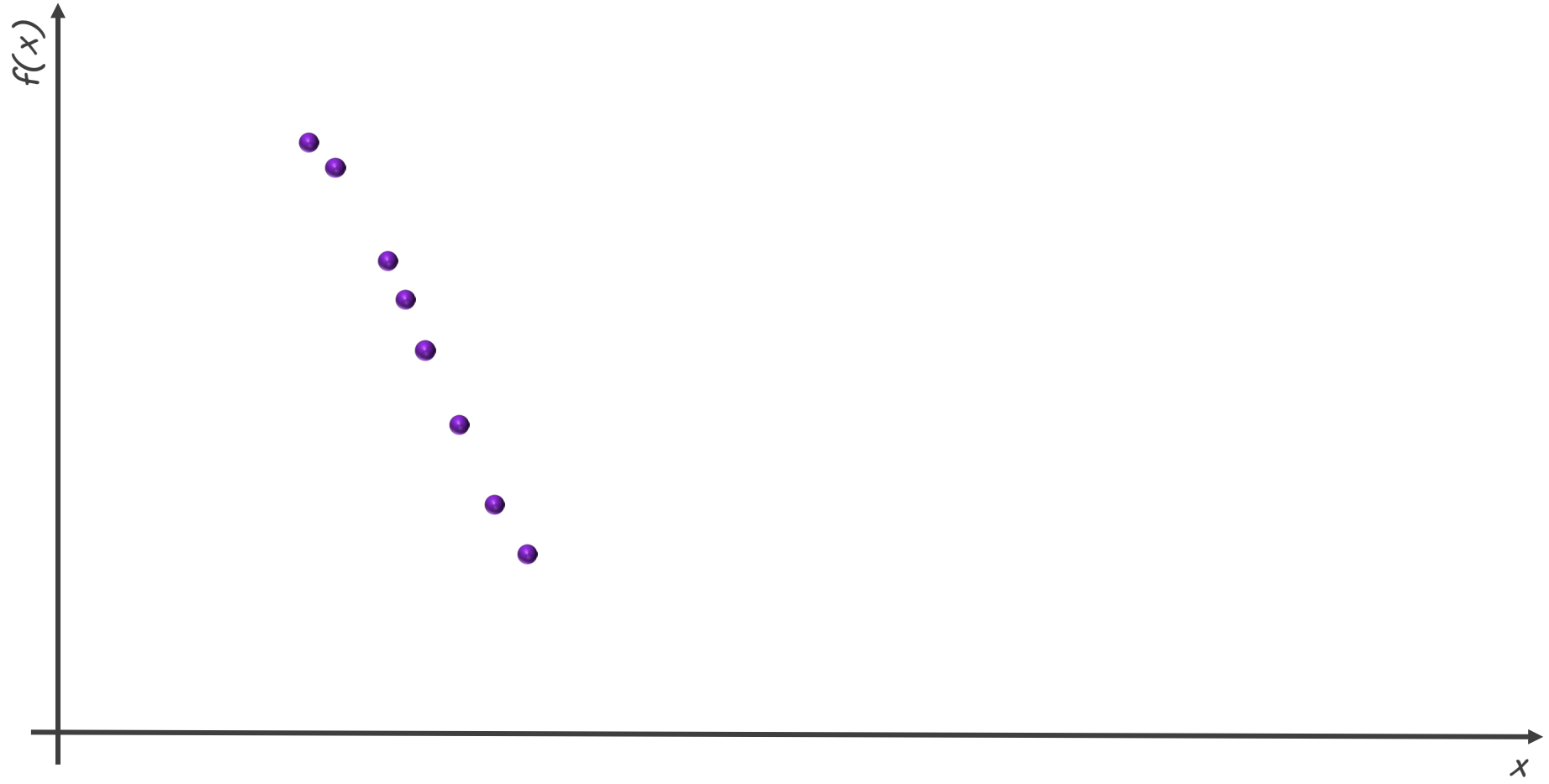
JÖNKÖPING UNIVERSITY



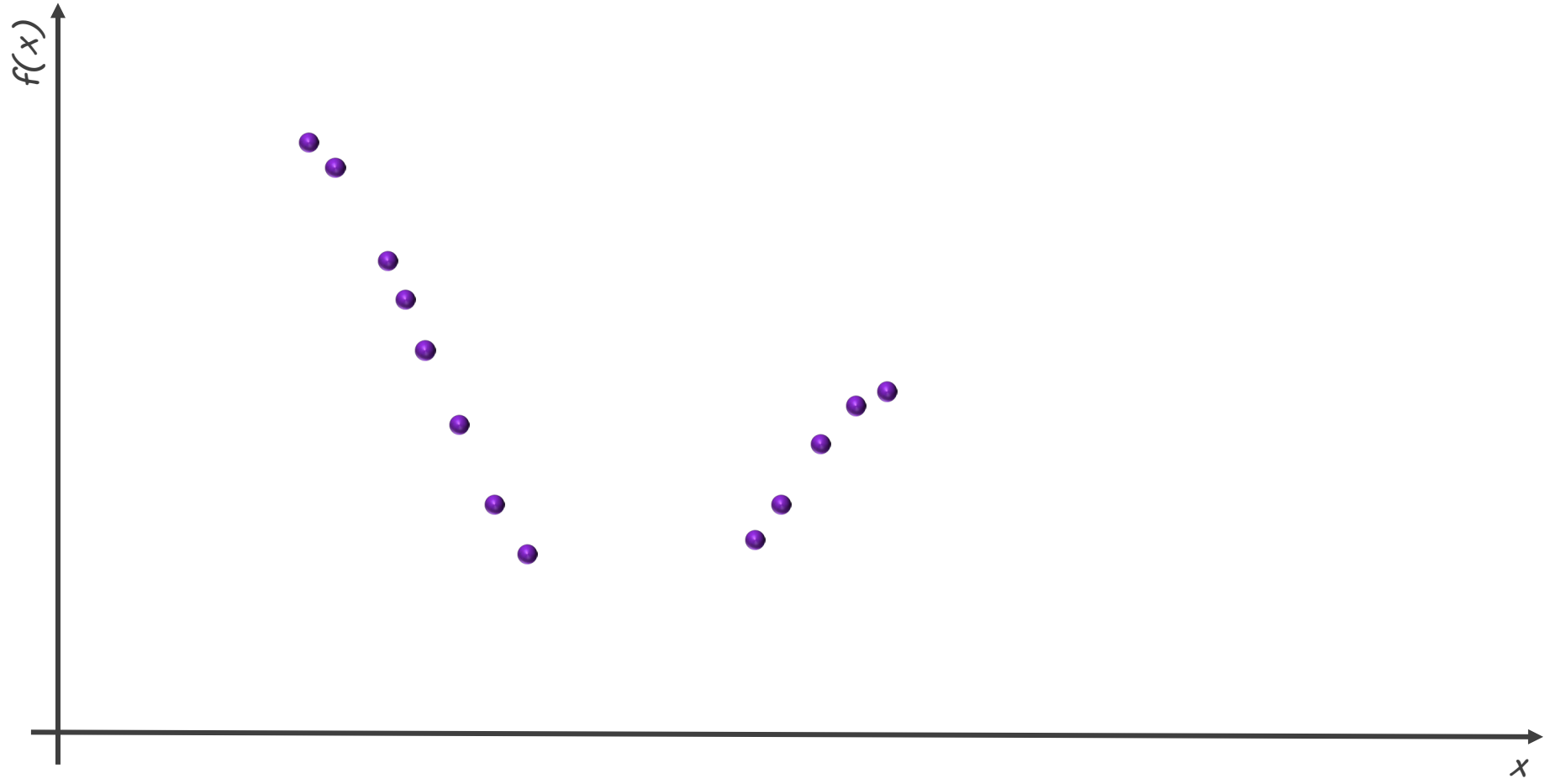


JÖNKÖPING UNIVERSITY

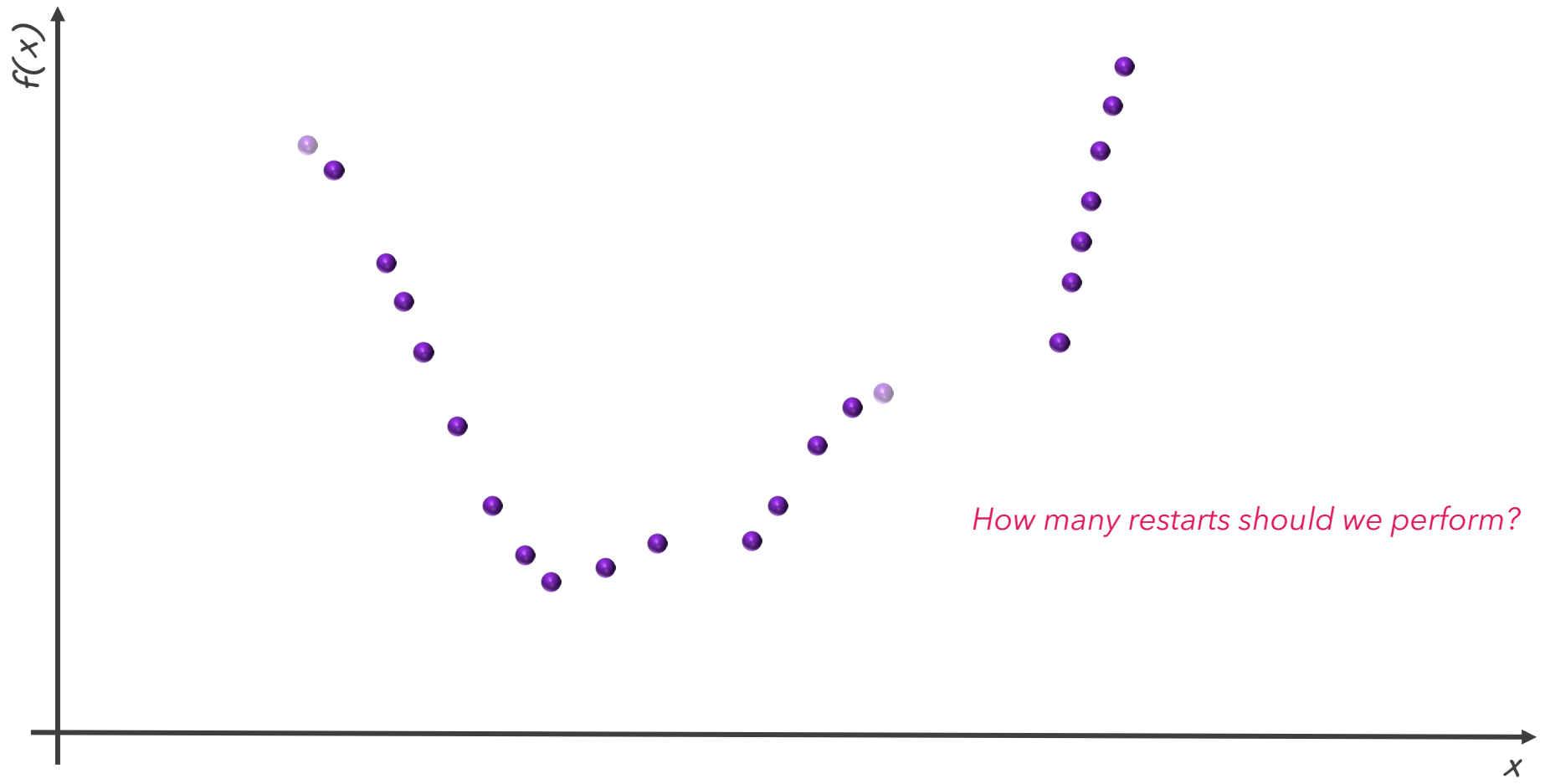
Random-restart Hill-Climbing



Random-restart Hill-Climbing



Random-restart Hill-Climbing



How many restarts should we perform?

Exploration-Exploitation trade-off

Exploration – generate candidate solutions in unexplored regions of the solution space

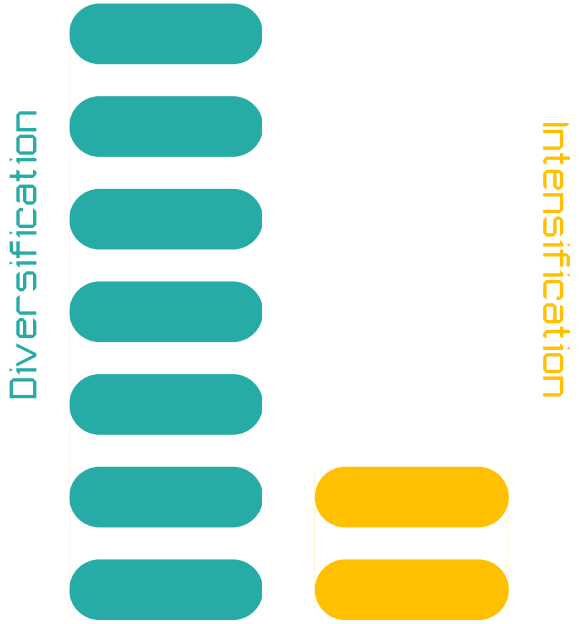
Diversification

Exploitation – search more intensively the promising regions of the solution space found so-far

Intensification

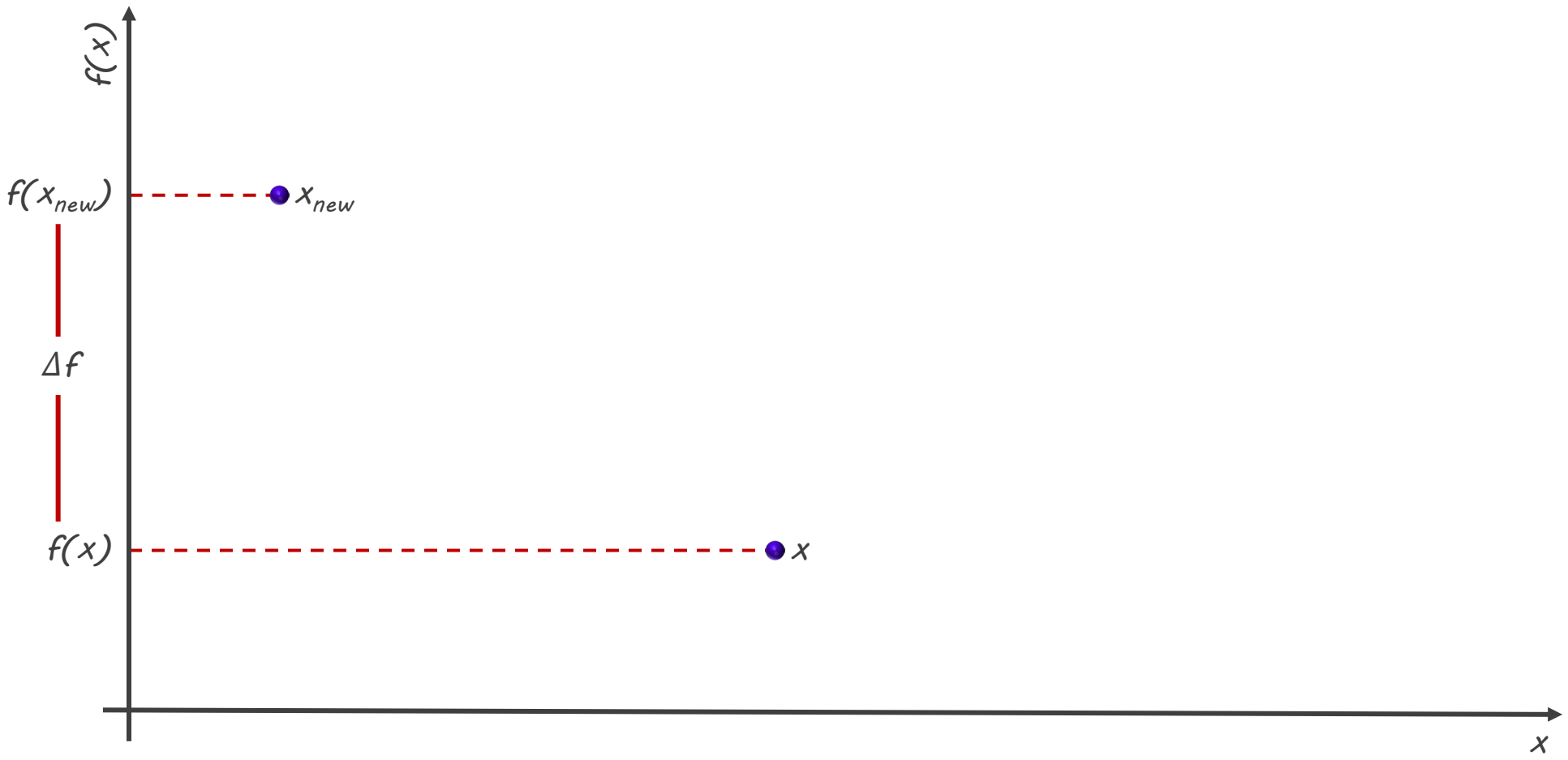
Exploration-Exploitation trade-off

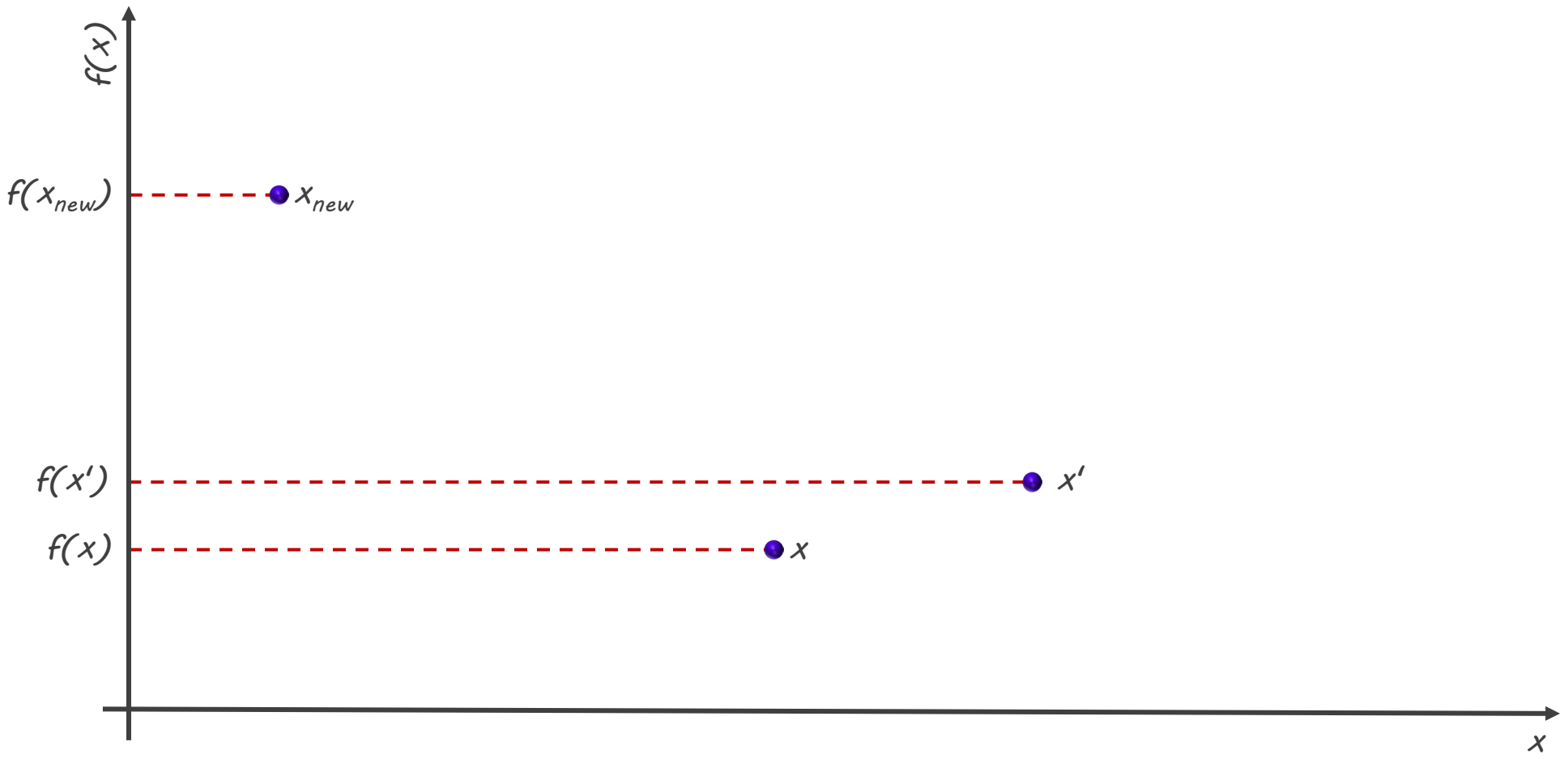
Exploration-Exploitation trade-off

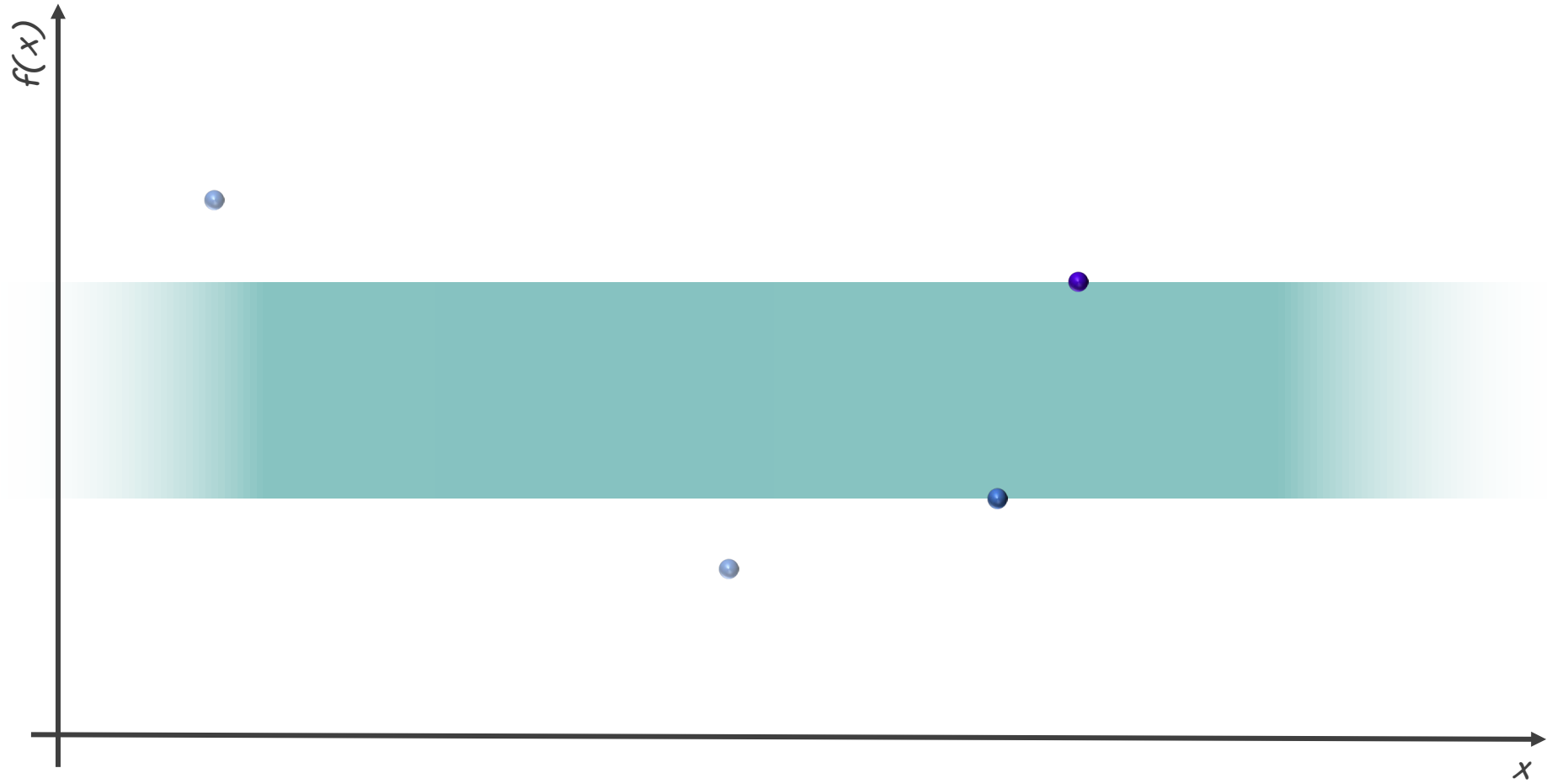


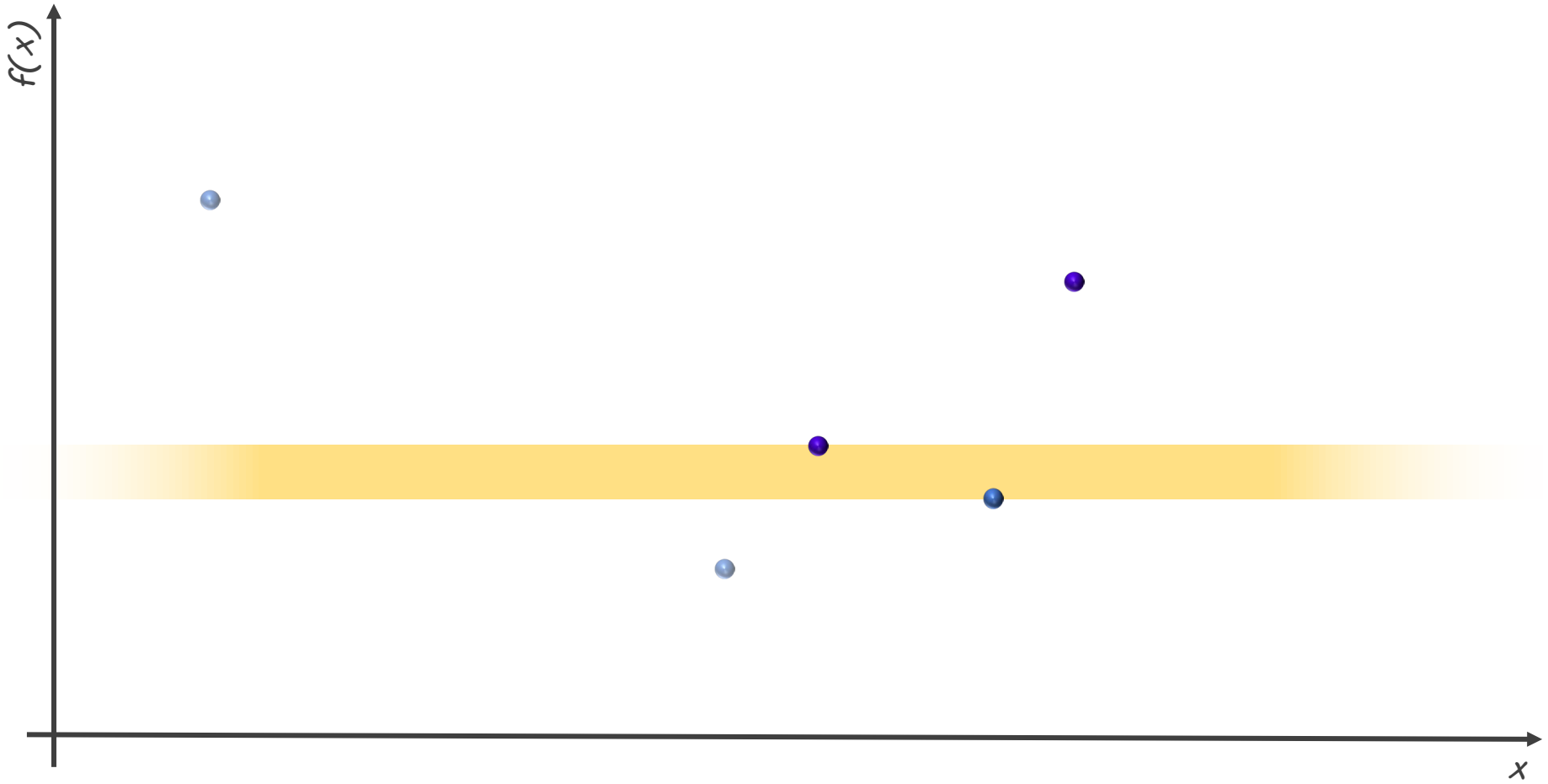
Exploration-Exploitation trade-off

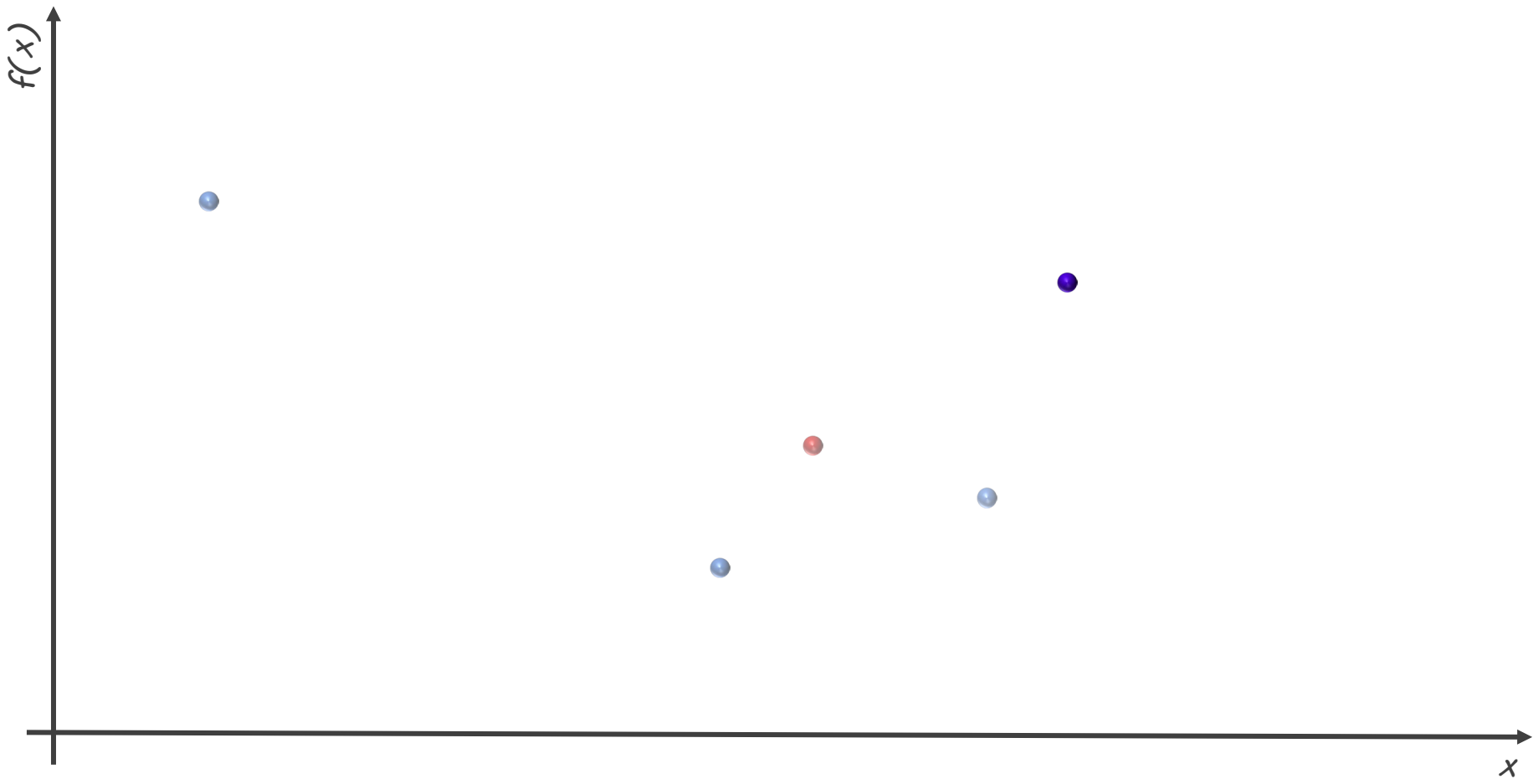
A commonly chosen trade-off is to
start with higher diversification (exploration) and
gradually shift toward higher intensification (exploitation).

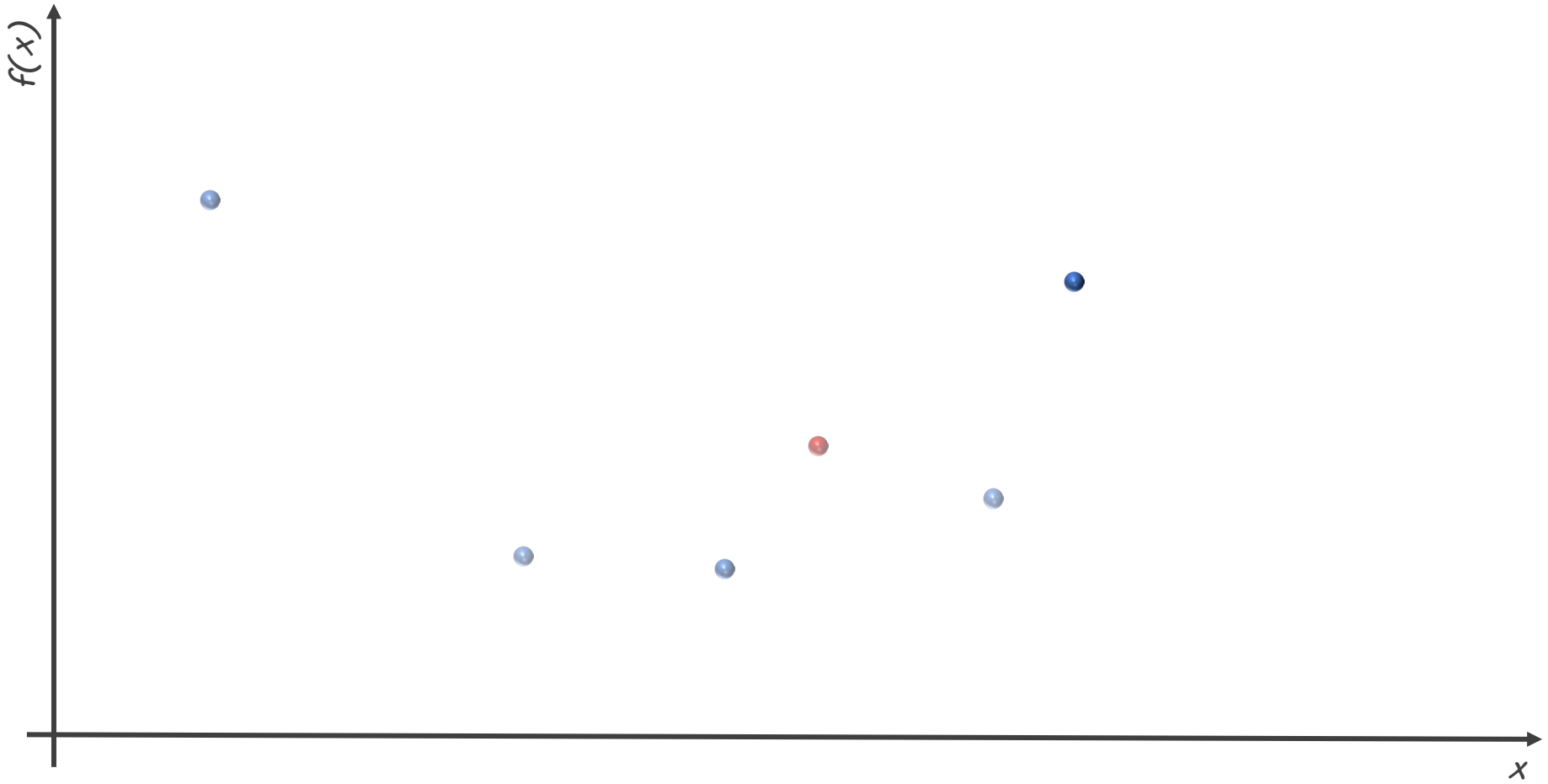

f(x)
f(x_new)
x_new
Δf
f(x)
x
x

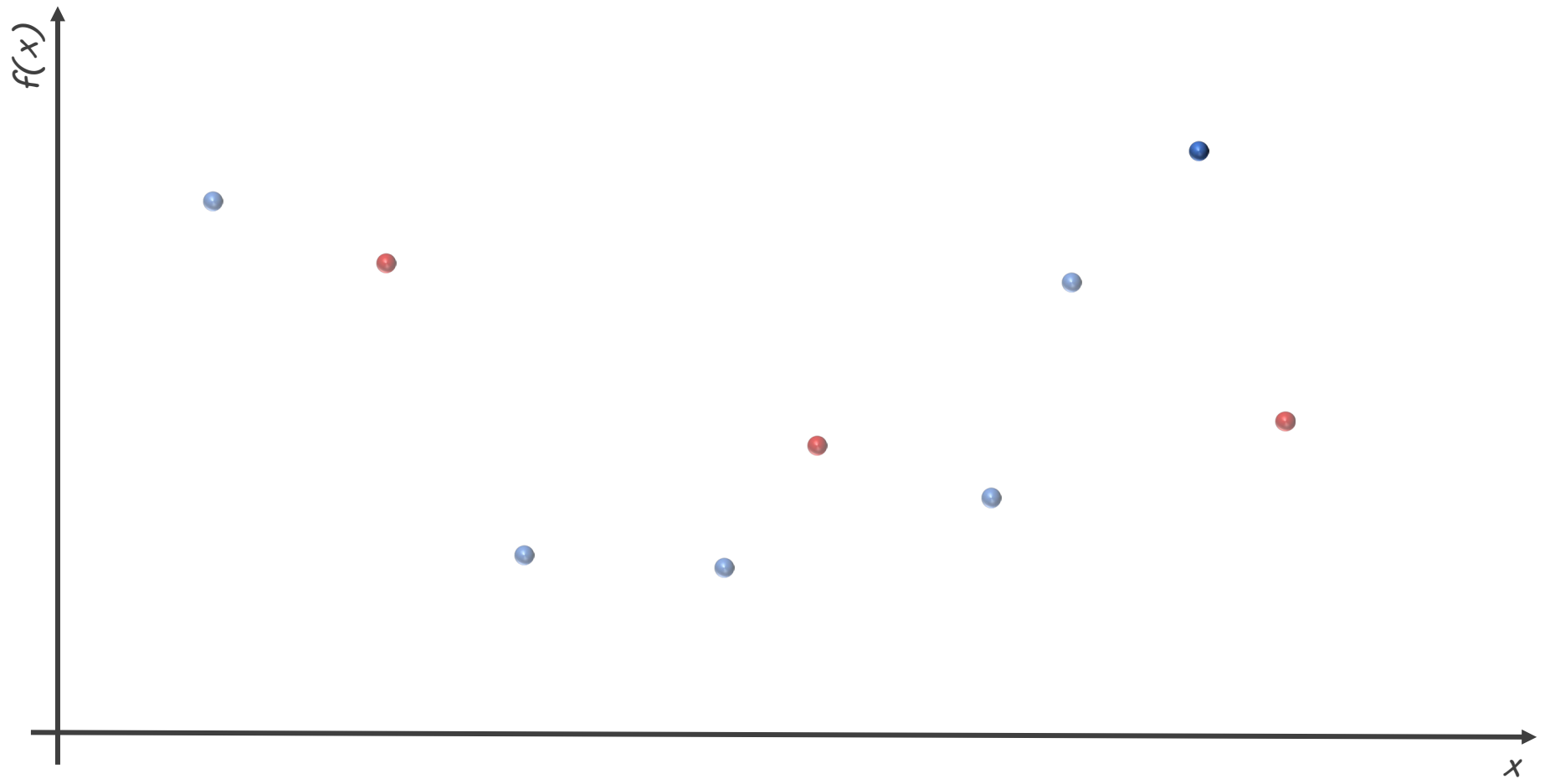

f(x)
f(x_new)
x_new
f(x')
x'
f(x)
x
x

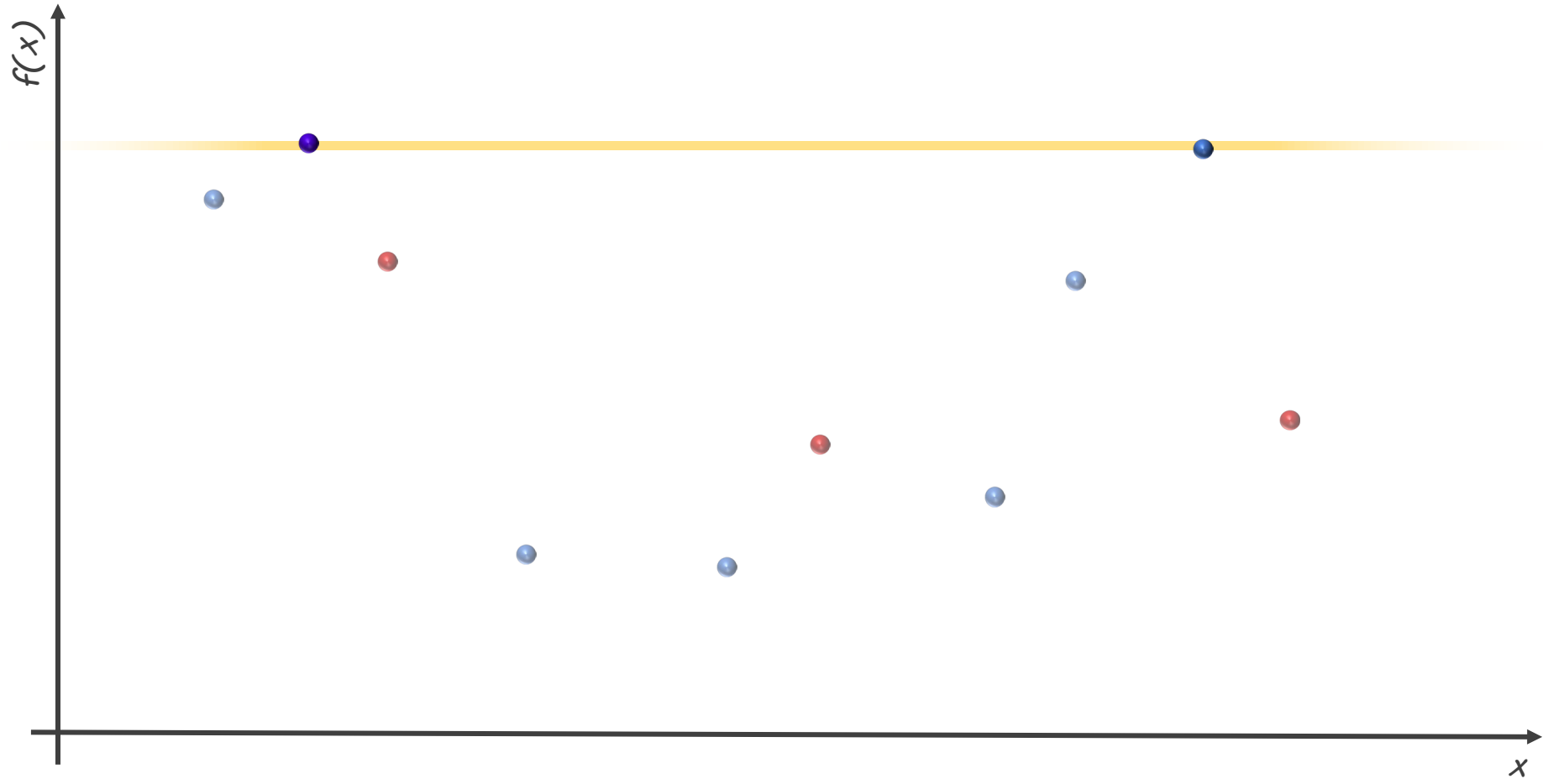

f(x)
x

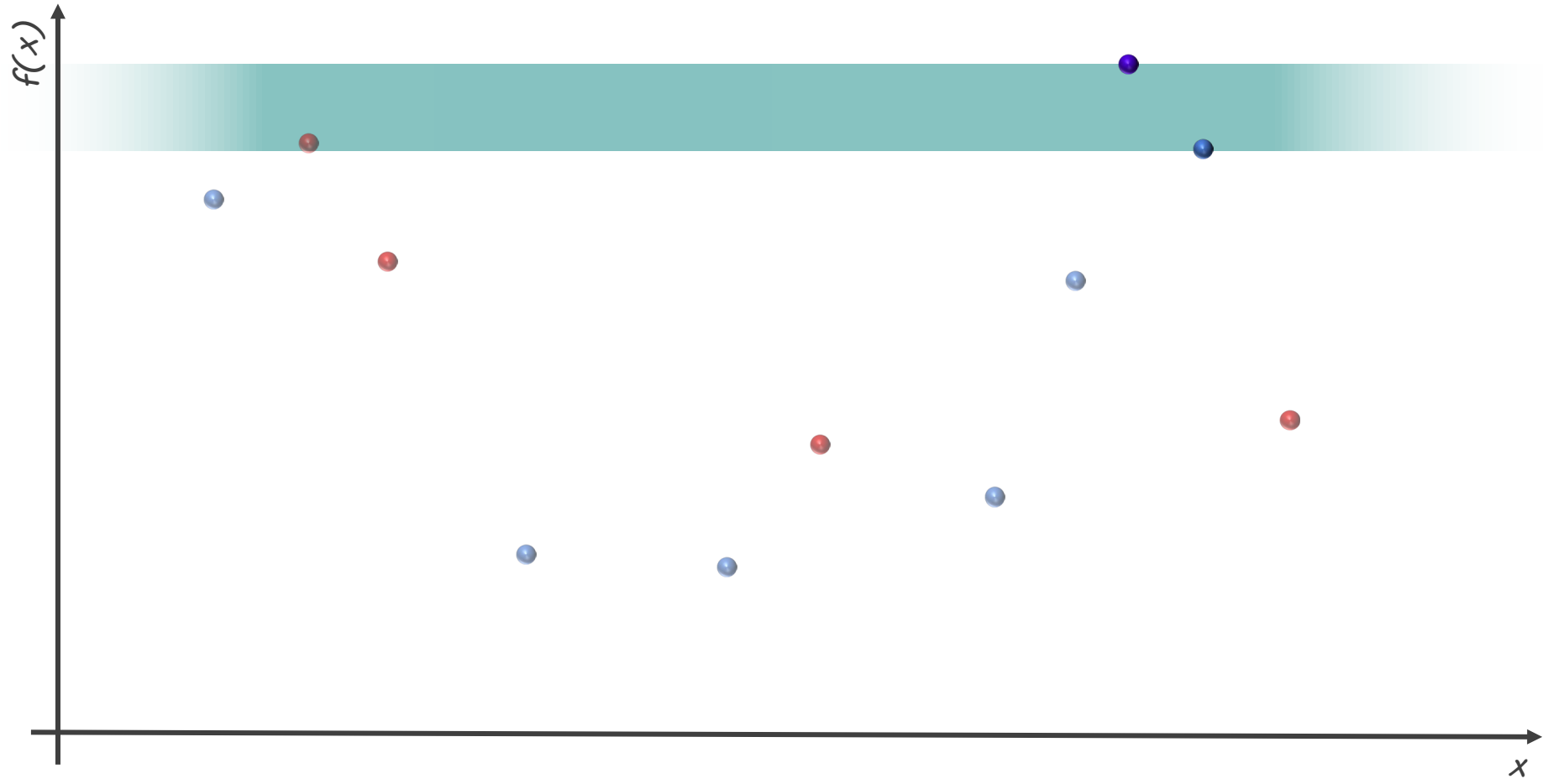

f(x)
x

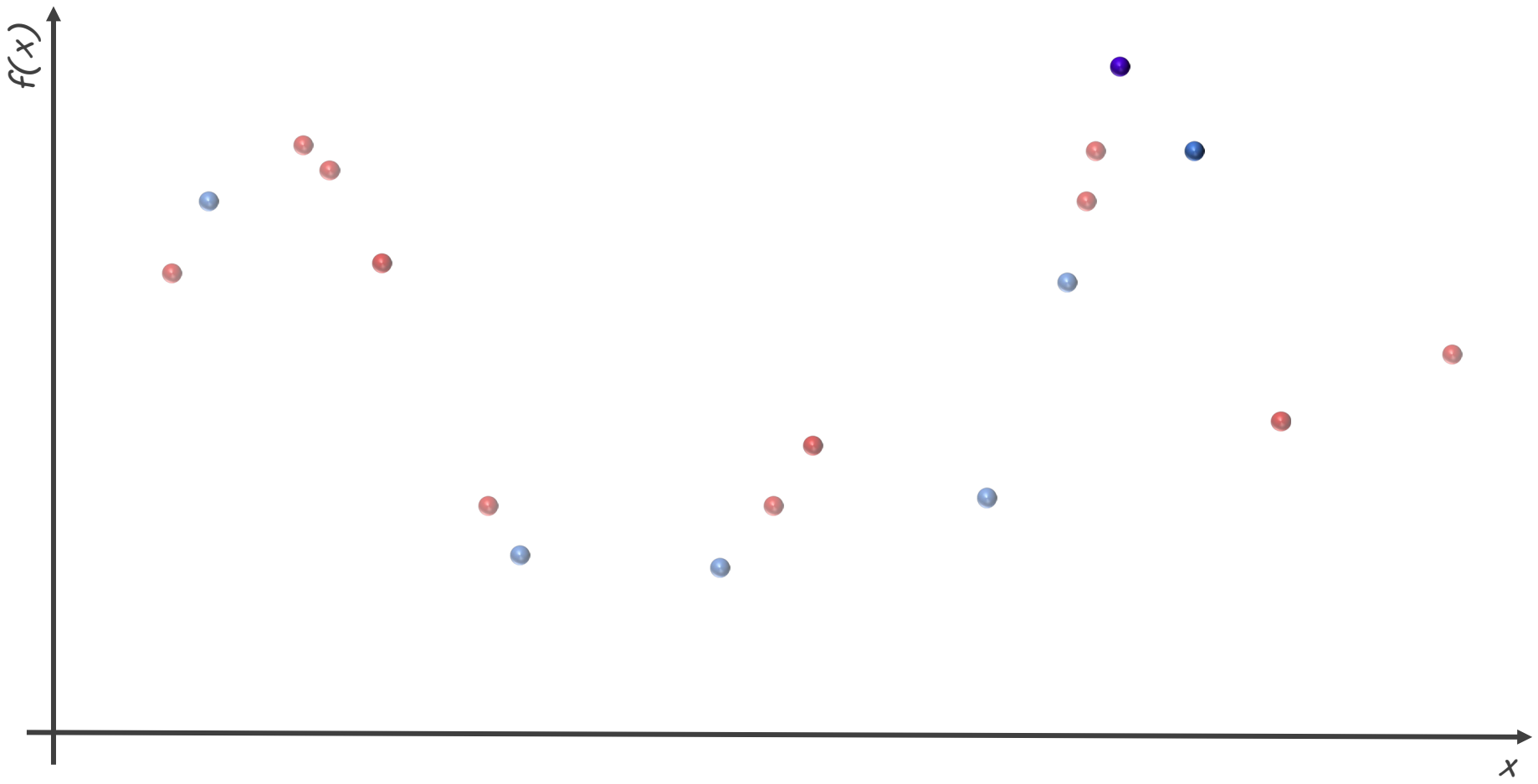

f(x)
x


f(x)
x


f(x)
x


f(x)
x


f(x)
x



Stochastic optimization

Trajectory-based algorithms

Dr. Alexandros Tzanetos

Simulated Annealing

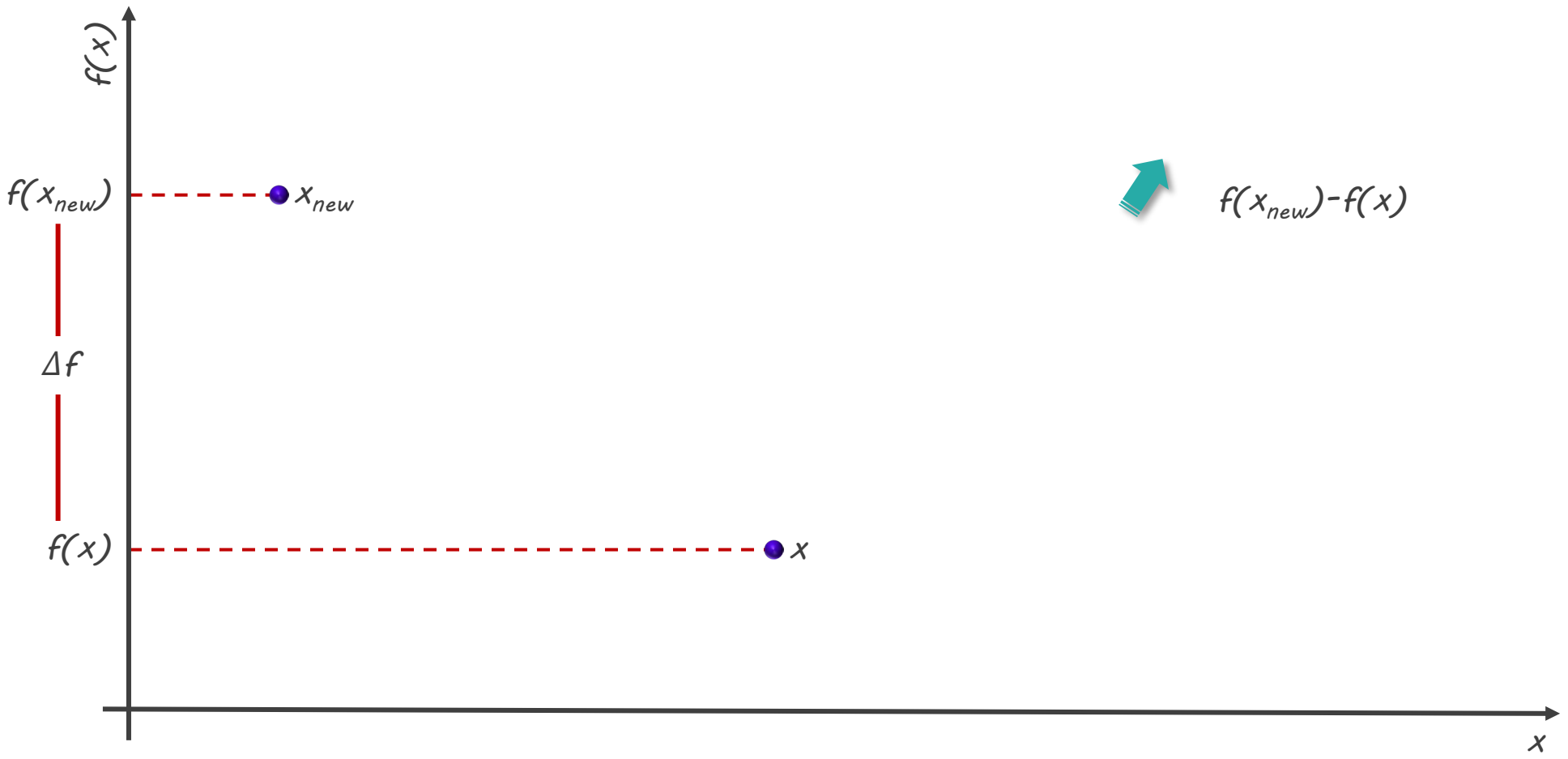
Annealing: the process used to temper or harden metals and glass by heating them to a high temperature and then gradually cooling them.

Simulated Annealing

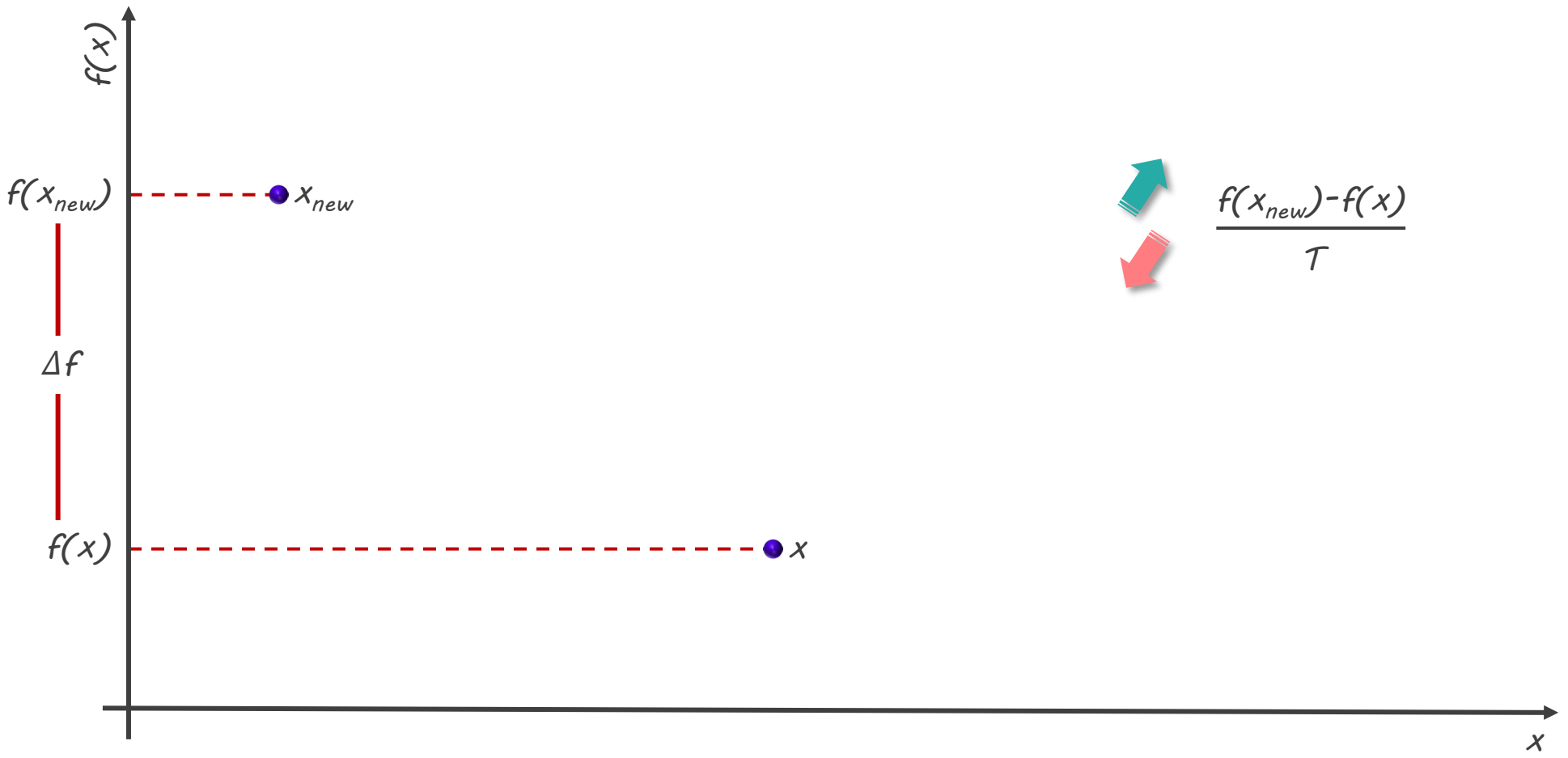
We define a parameter called Temperature to control the trade-off.

- ❑ early stages ⇨ more explorative
- ❑ later on ⇨ more exploitative

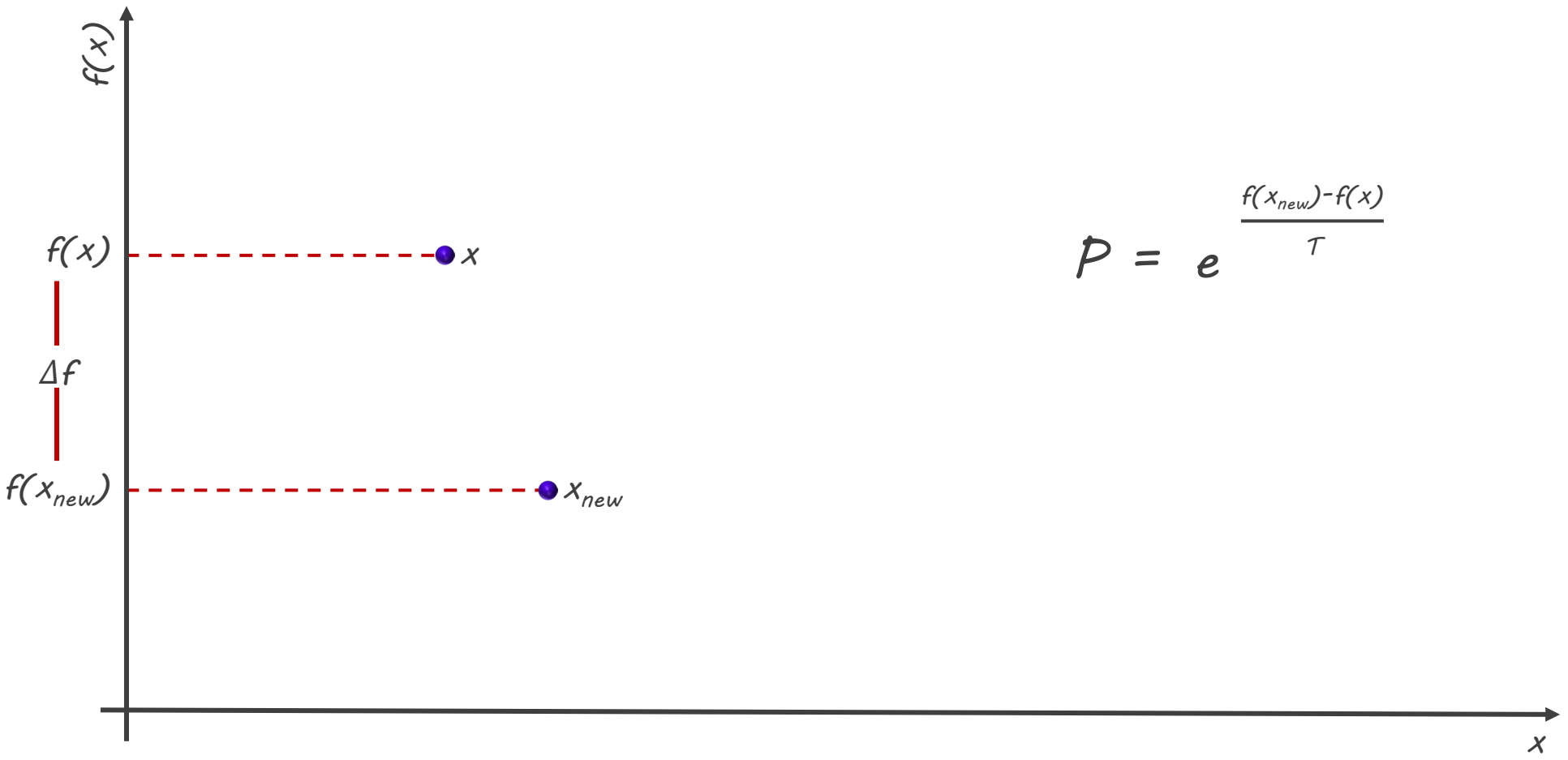
Simulated Annealing



Simulated Annealing



Simulated Annealing



$$P = e^{\frac{f(x_{new})-f(x)}{T}}$$

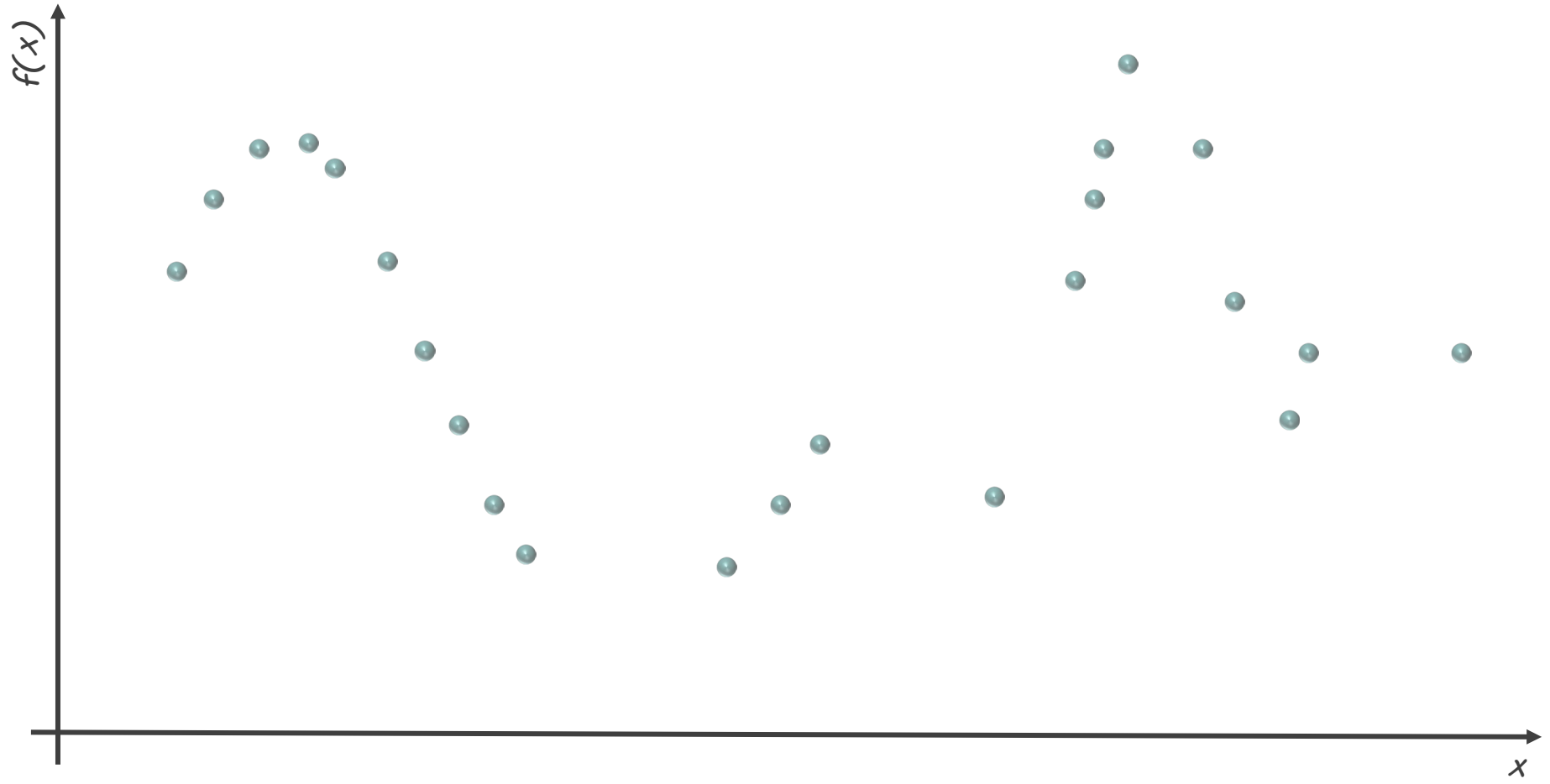
Tabu Search

We create a list of some recently seen candidate solutions called Tabu List to avoid revisiting them.

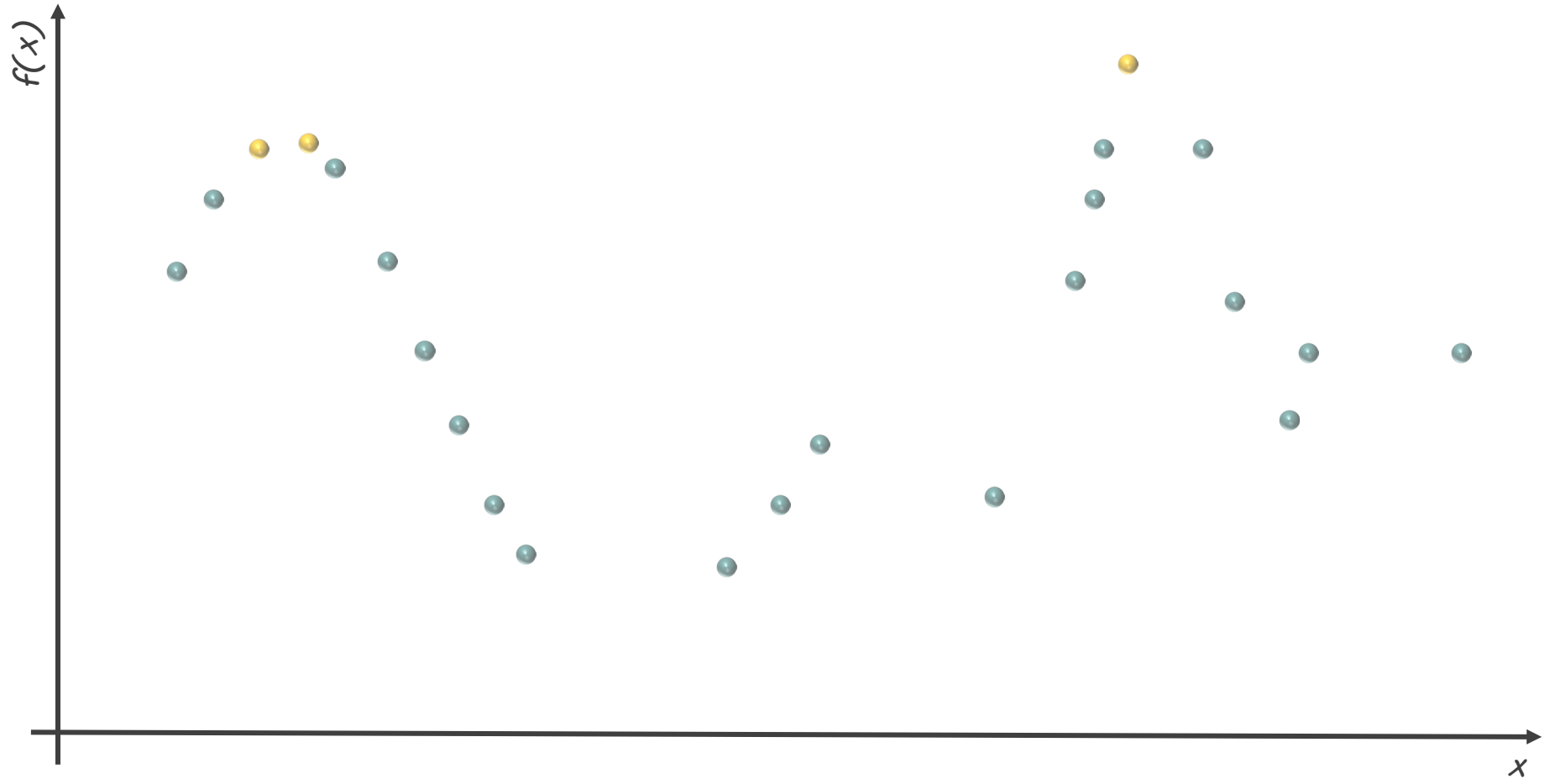
…but it looks like this works only for discrete problems

One solution is to avoid "similar" solutions by using a similarity distance measure.

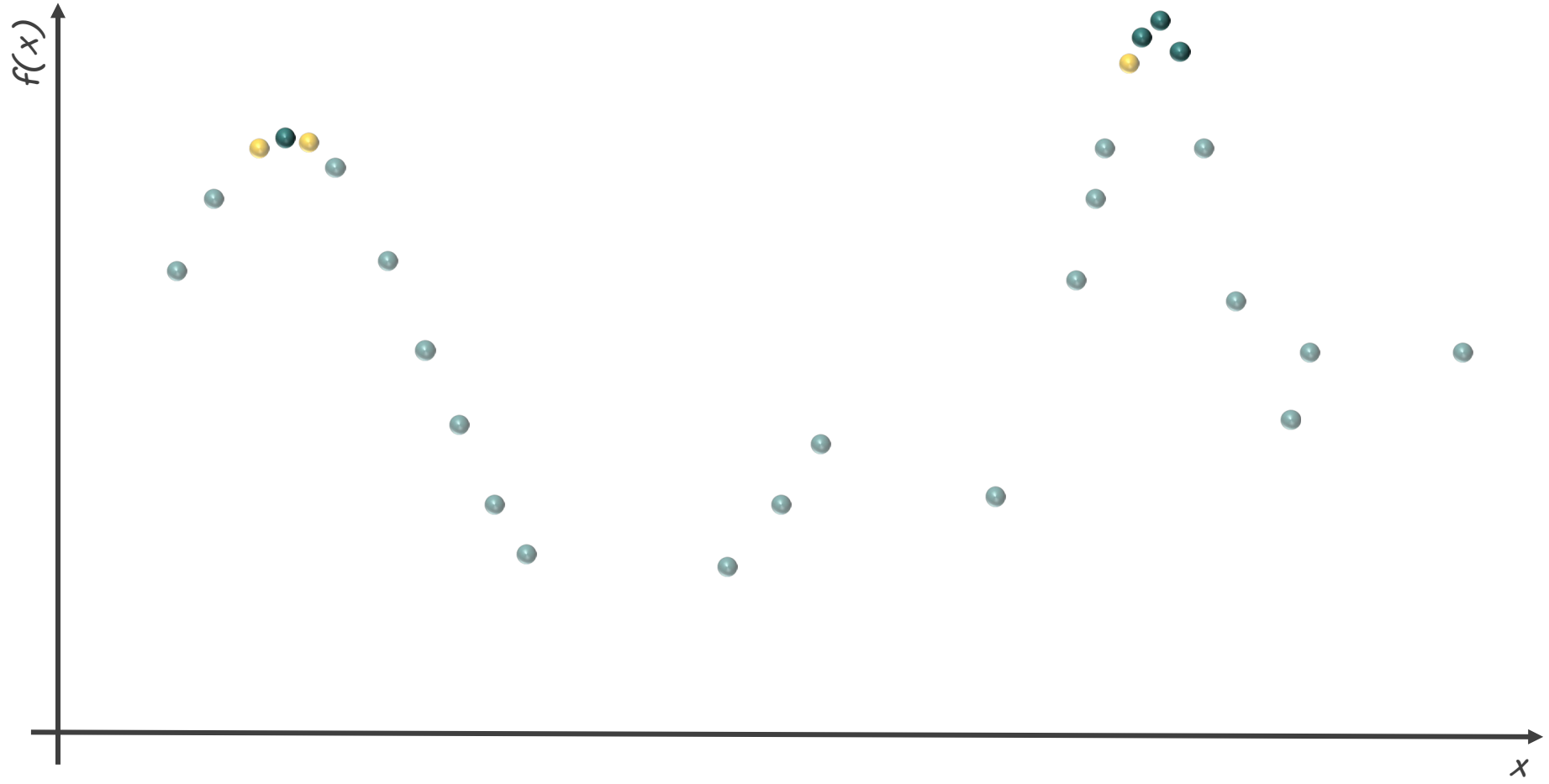
Exploiting the local optima



Exploiting the local optima

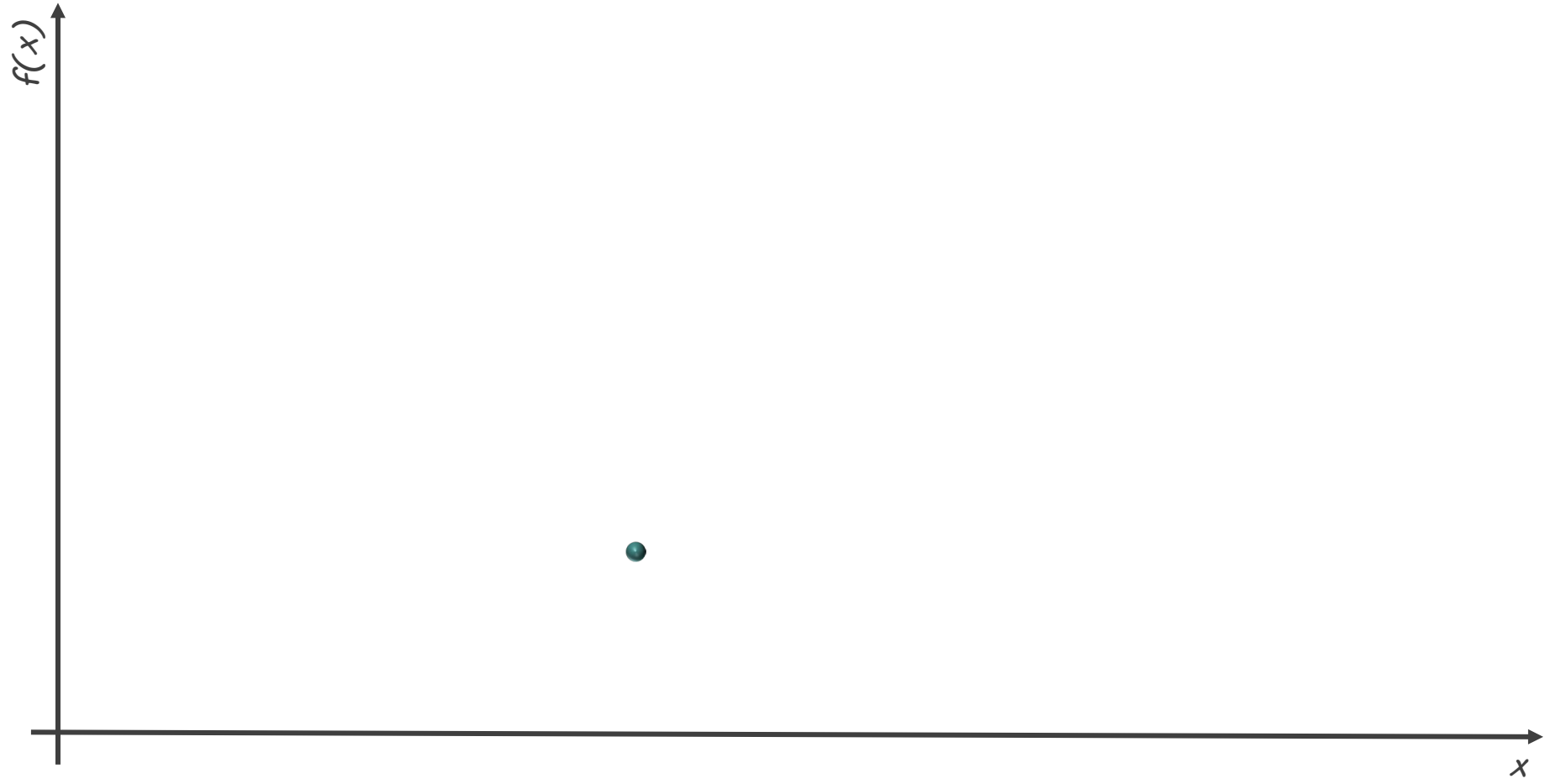


Exploiting the local optima

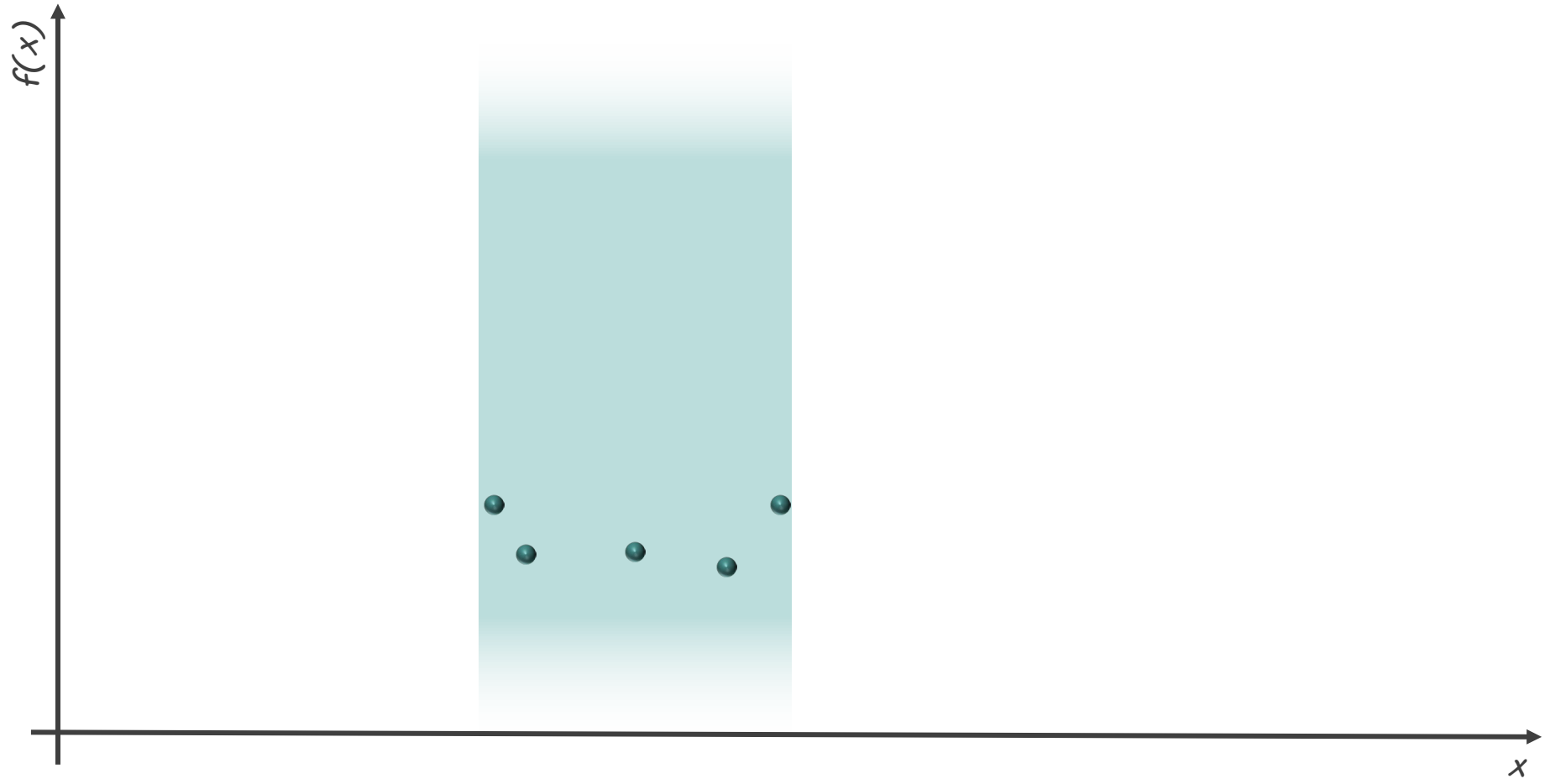


Iterated Local Search (ILS)

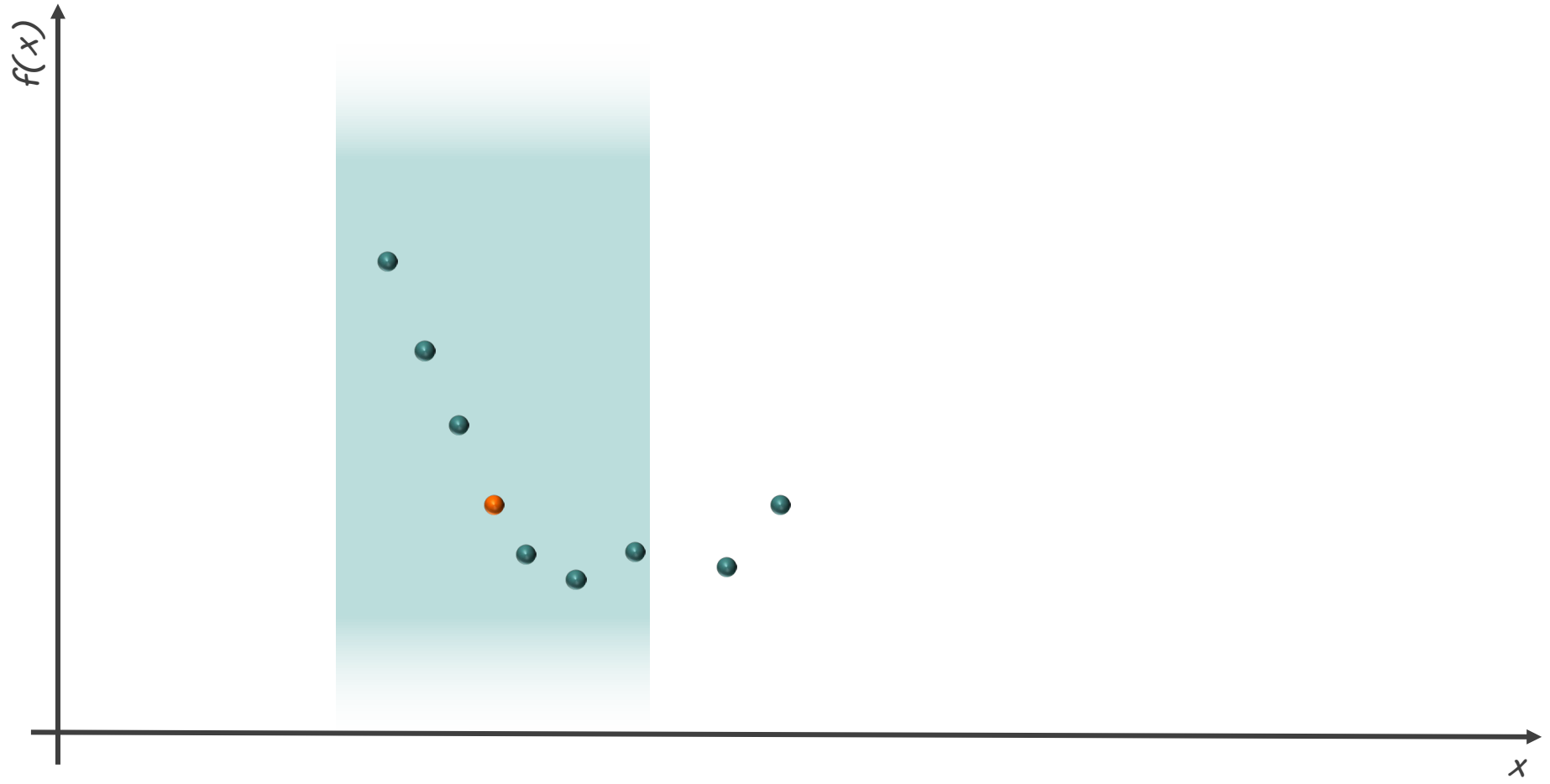
Iterated Local Search (ILS)



Iterated Local Search (ILS)



Iterated Local Search (ILS)



Population-based algorithms

Dr. Alexandros Tzanetos

Working with a population of solutions

Instead of creating and tweaking one solution, we can generate a population of solutions.

- Evolutionary Algorithms (EAs)
- Swarm Intelligence
- …
- Evolution Strategies (ES)

Evolution Strategies: an Evolutionary Computing paradigm where a predefined number of parents is selected to generate a predefined number of offspring.

Evolution Strategies (ES)

μ: number of selected parents

λ: number of children generated by the parents

Two variants exist:

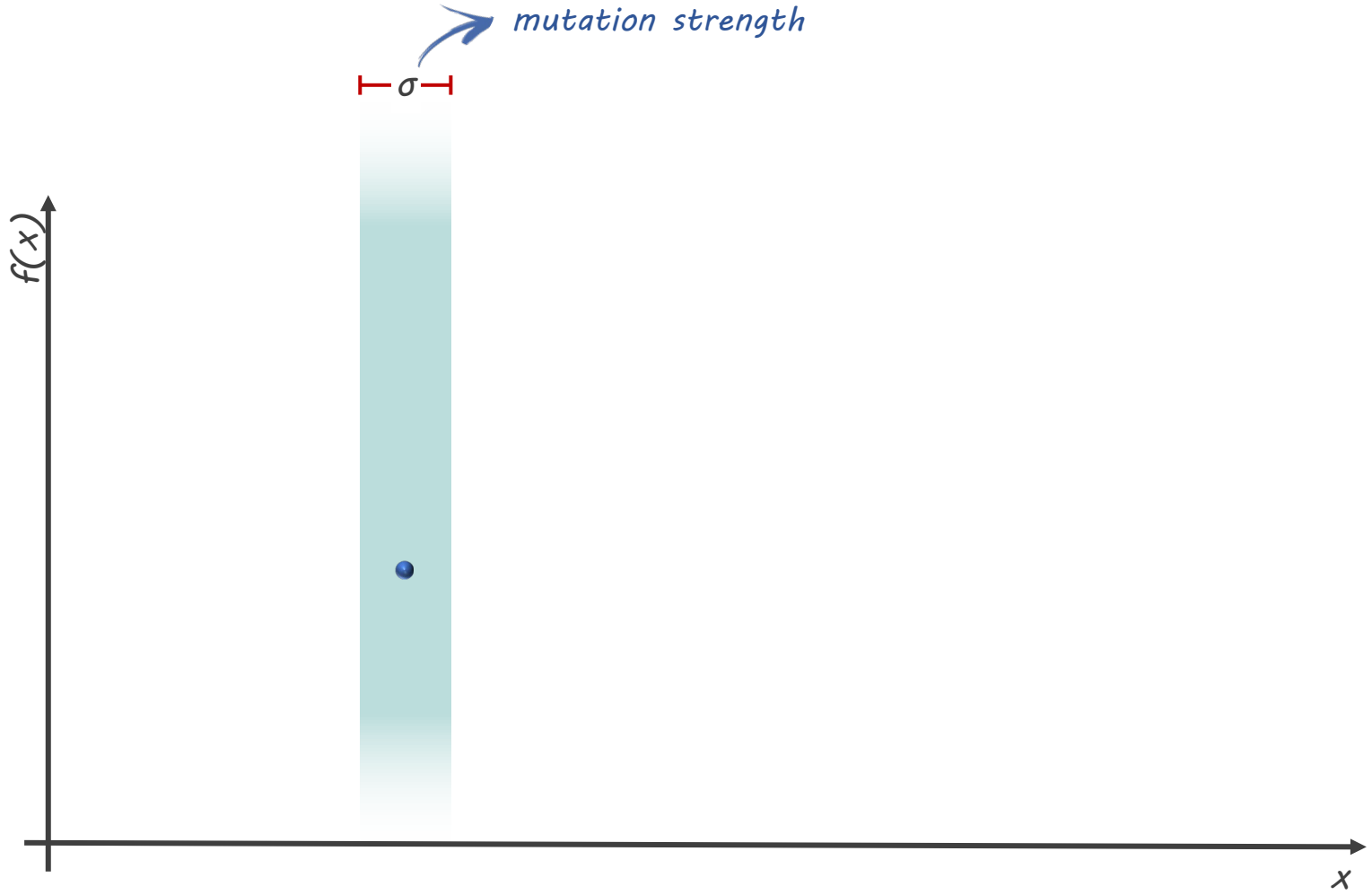
- (μ, λ) the μ best candidate solutions create λ offspring which replace all parents
- $(\mu + \lambda)$ the best offspring are selected from a union of μ parents and λ offspring

$(1 + 1)$**ES**

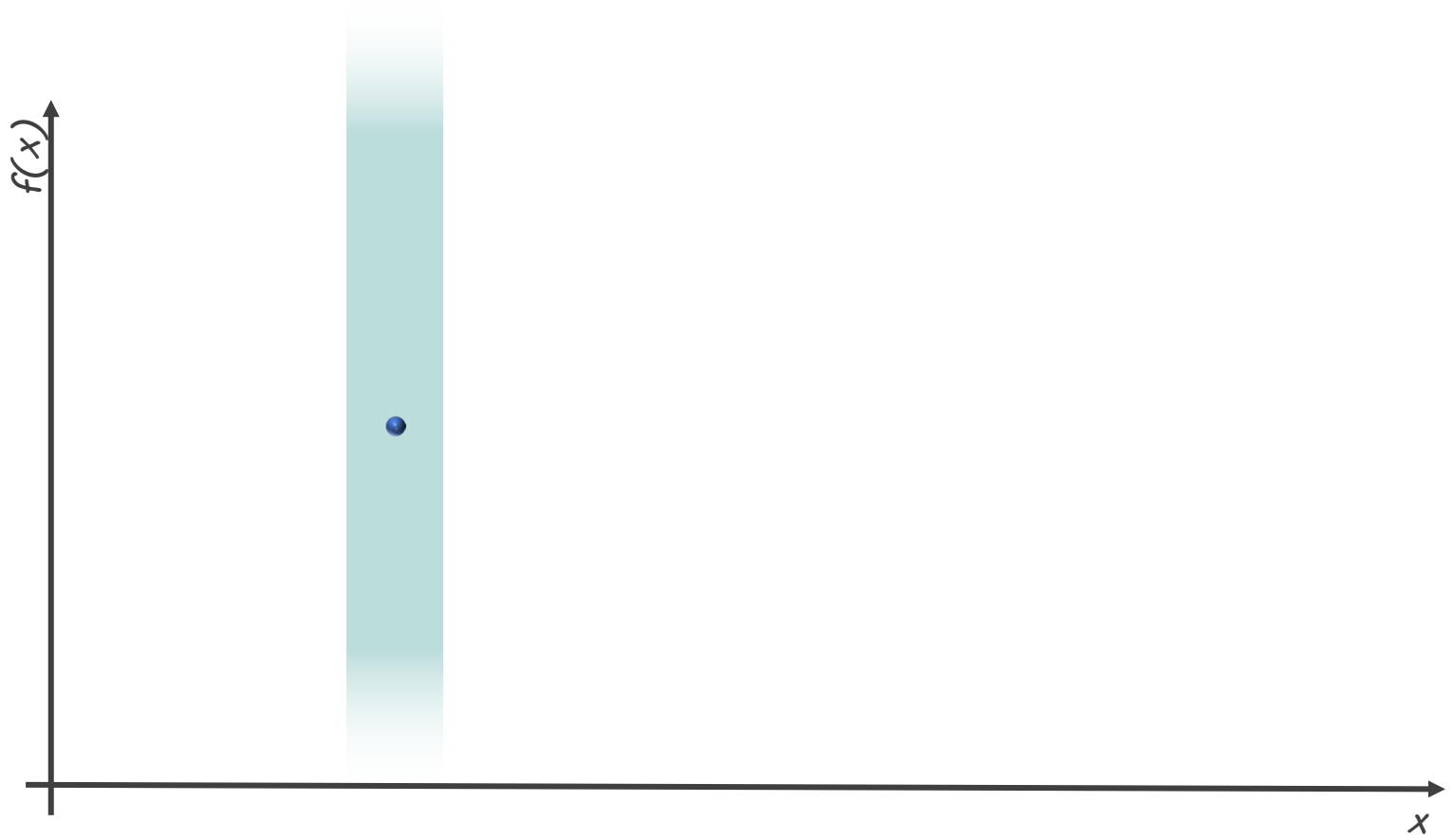
…the simplest ES

- population size: 2
- evolution: the best candidate solution generates 1 offspring
- selection: if the offspring is better than the parent, it replaces it

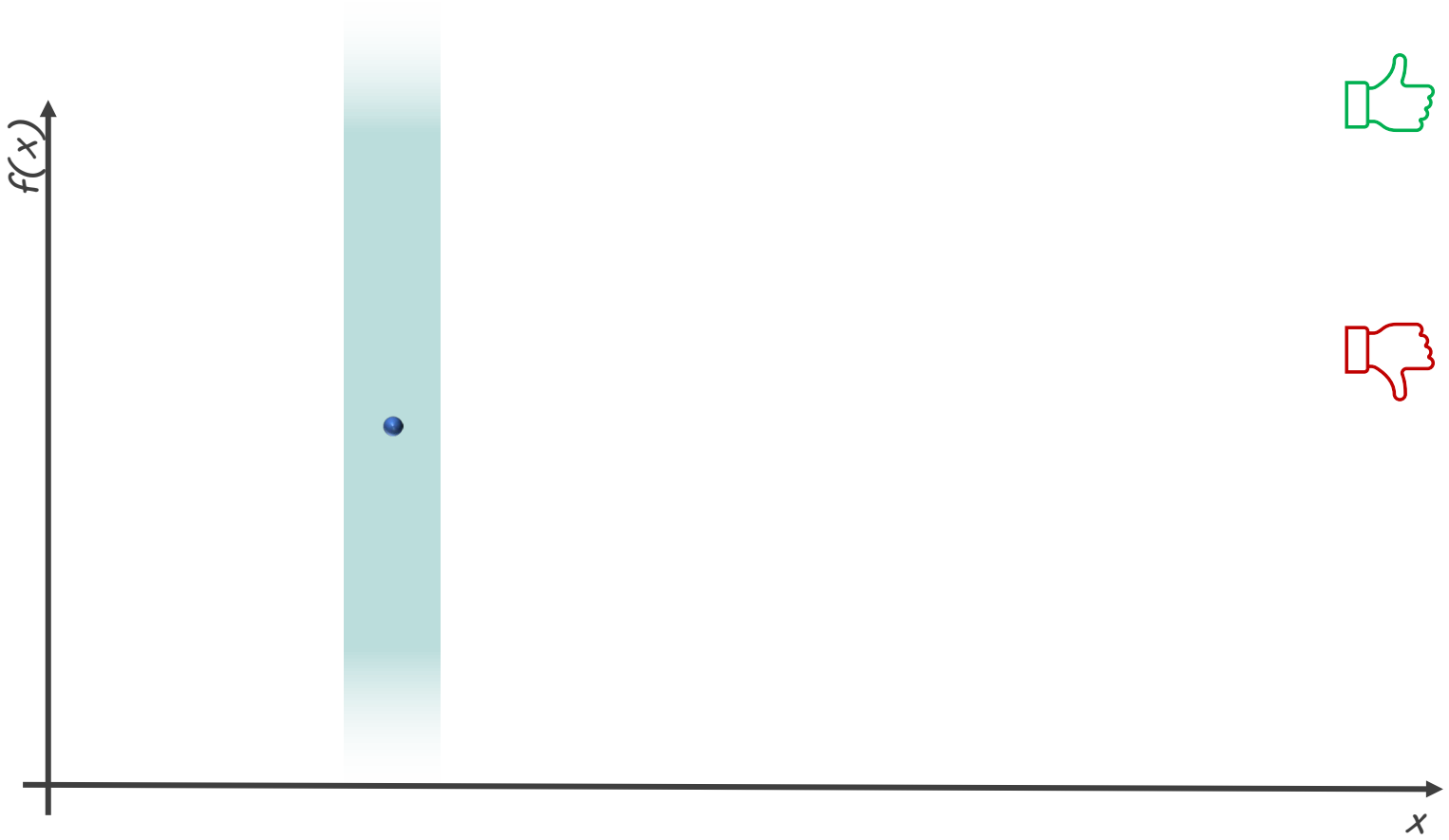
$(1+1)$**ES**



$(1+1)$**ES**



$(1+1)$**ES**



great for local search

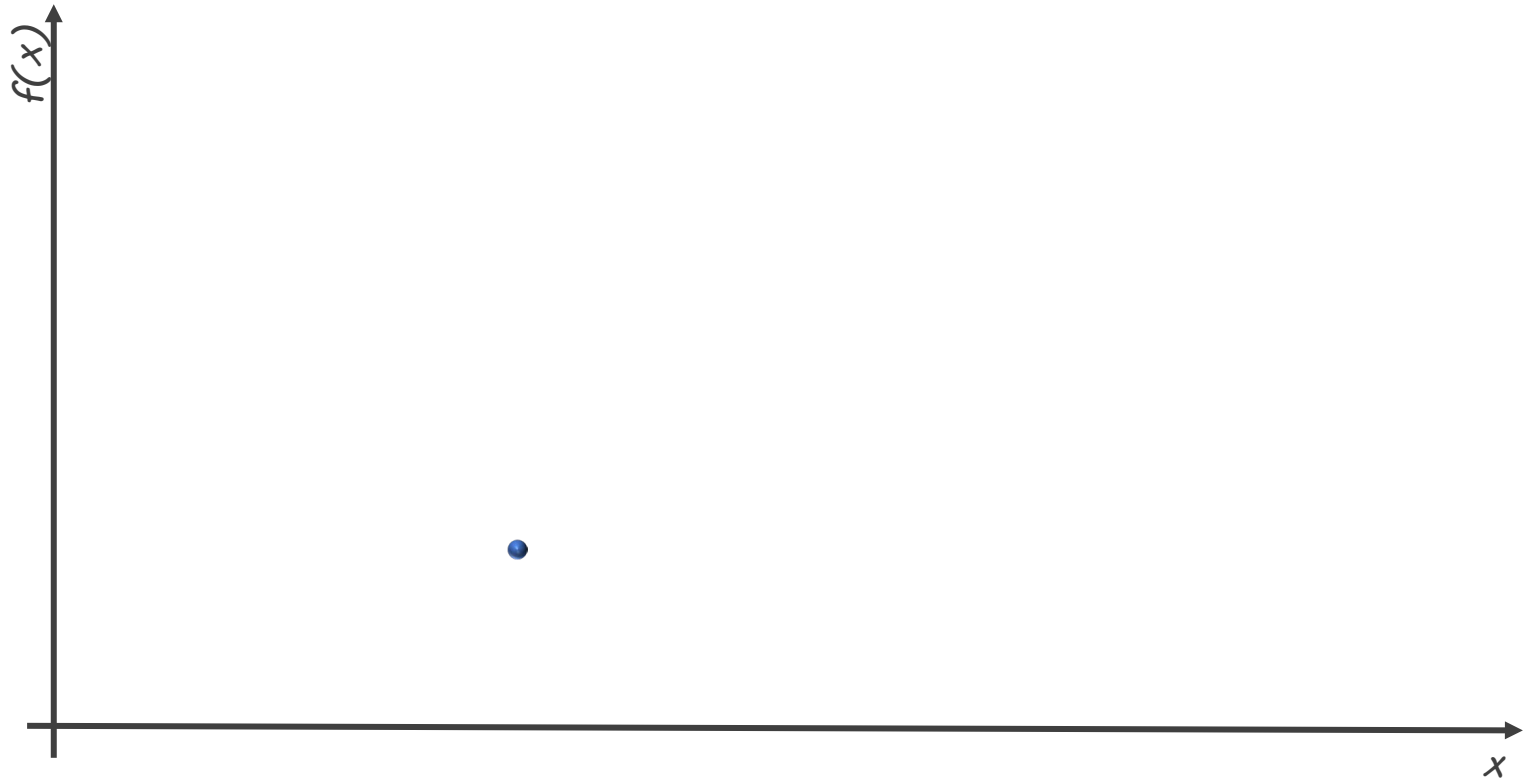
mainly exploitative

slow converging

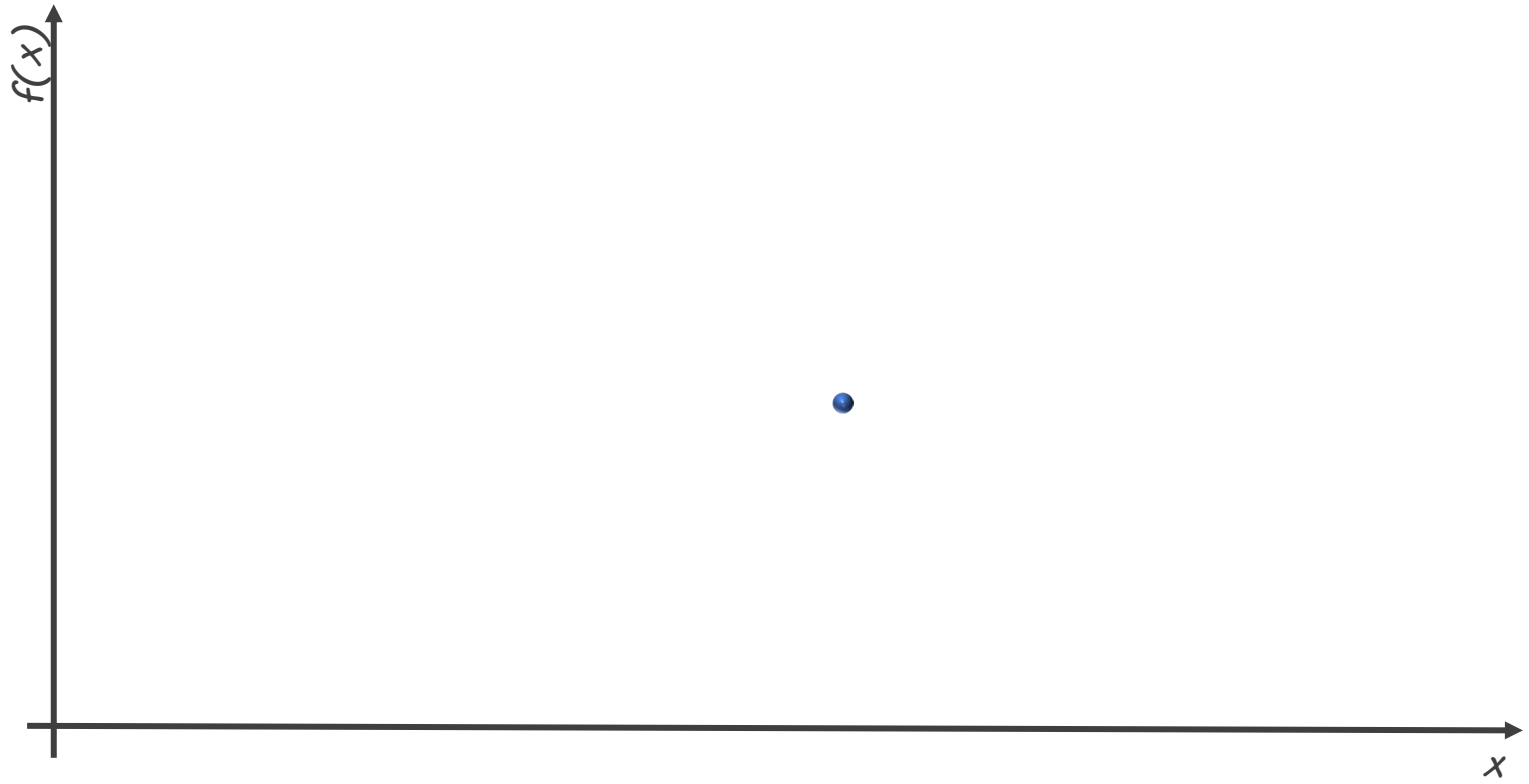
(1,1)**ES**

- population size: 2
- evolution: the best candidate solution generates 1 offspring
- selection: the offspring always replaces the parent

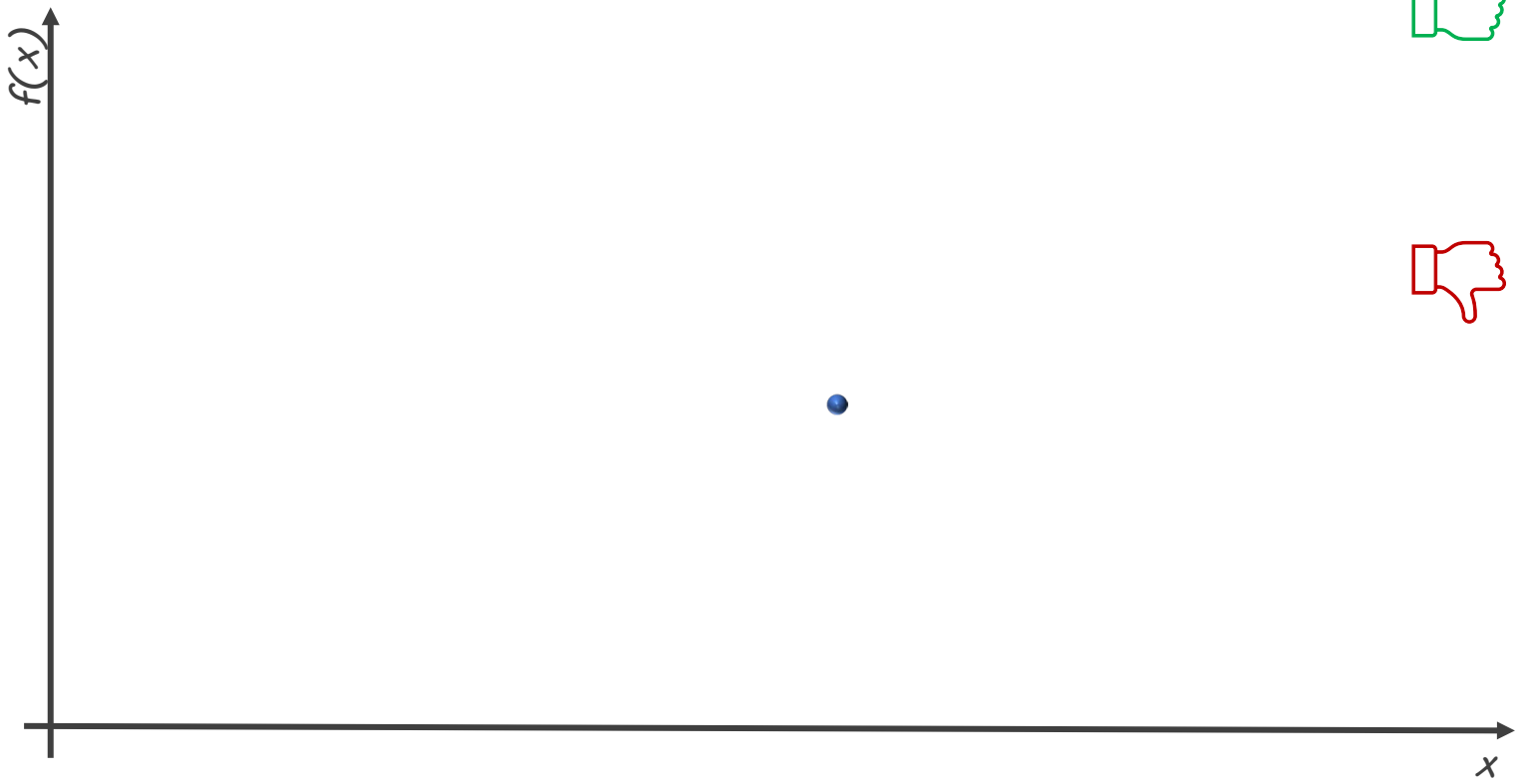
(1,1)**ES**



(1,1)**ES**



(1,1)**ES**



great for local search

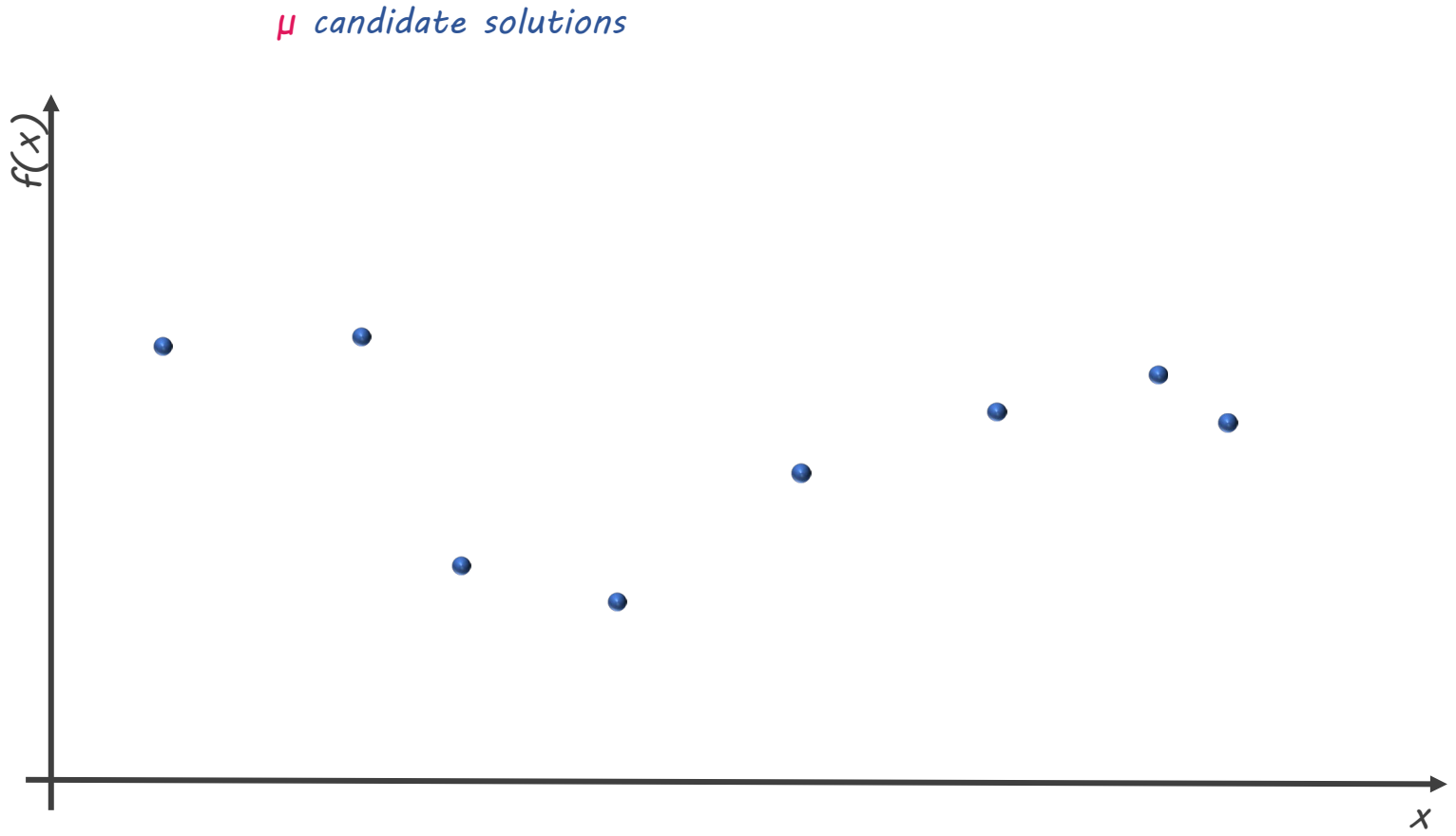
mainly exploitative

slow converging

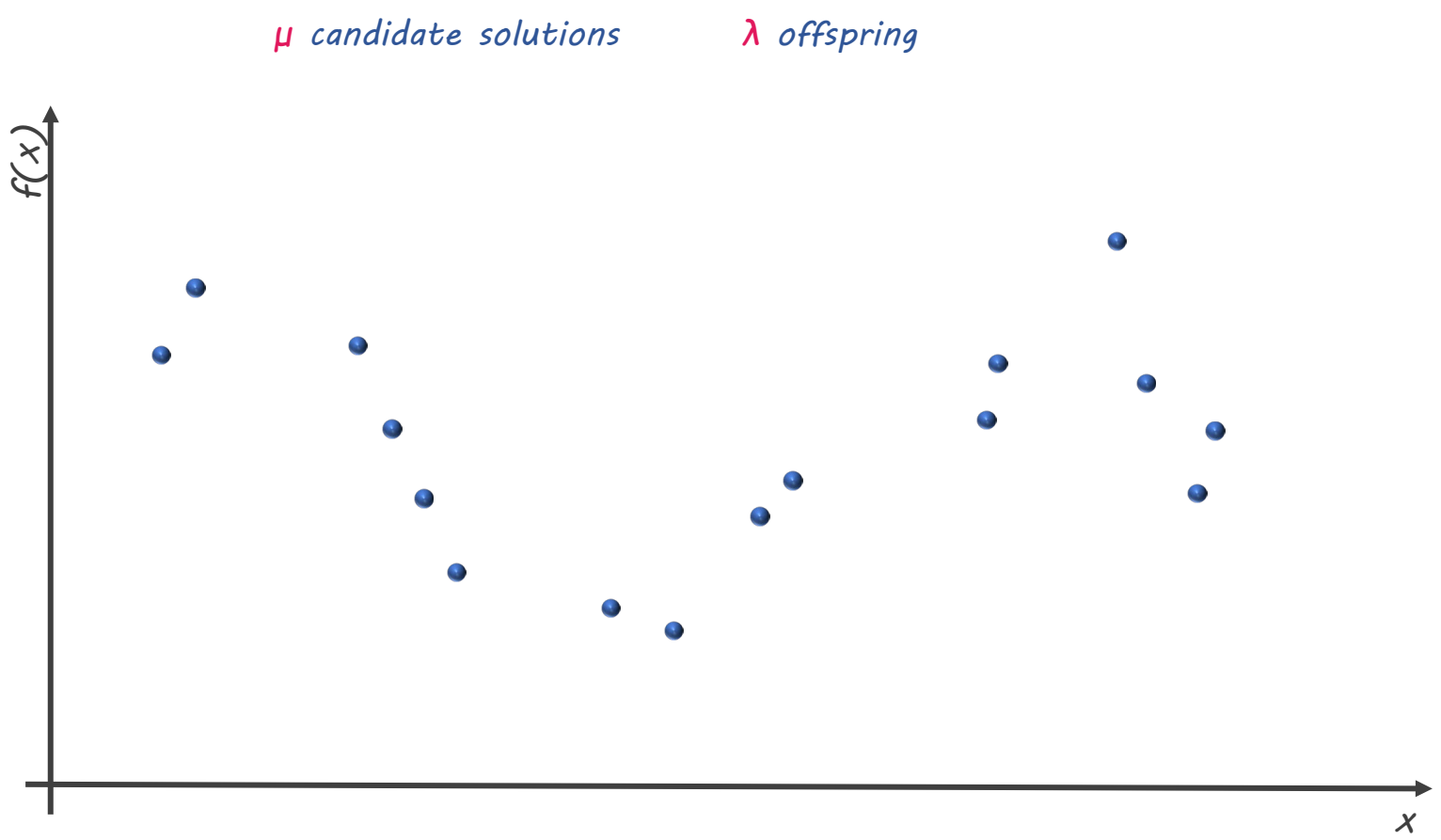
$(\mu + \lambda)$ES

- population size: $\mu + \lambda$
- evolution: the μ best candidate solutions generate λ offspring
- selection: $\mu + \lambda$ solutions compete for the μ of them that contribute to the new generation

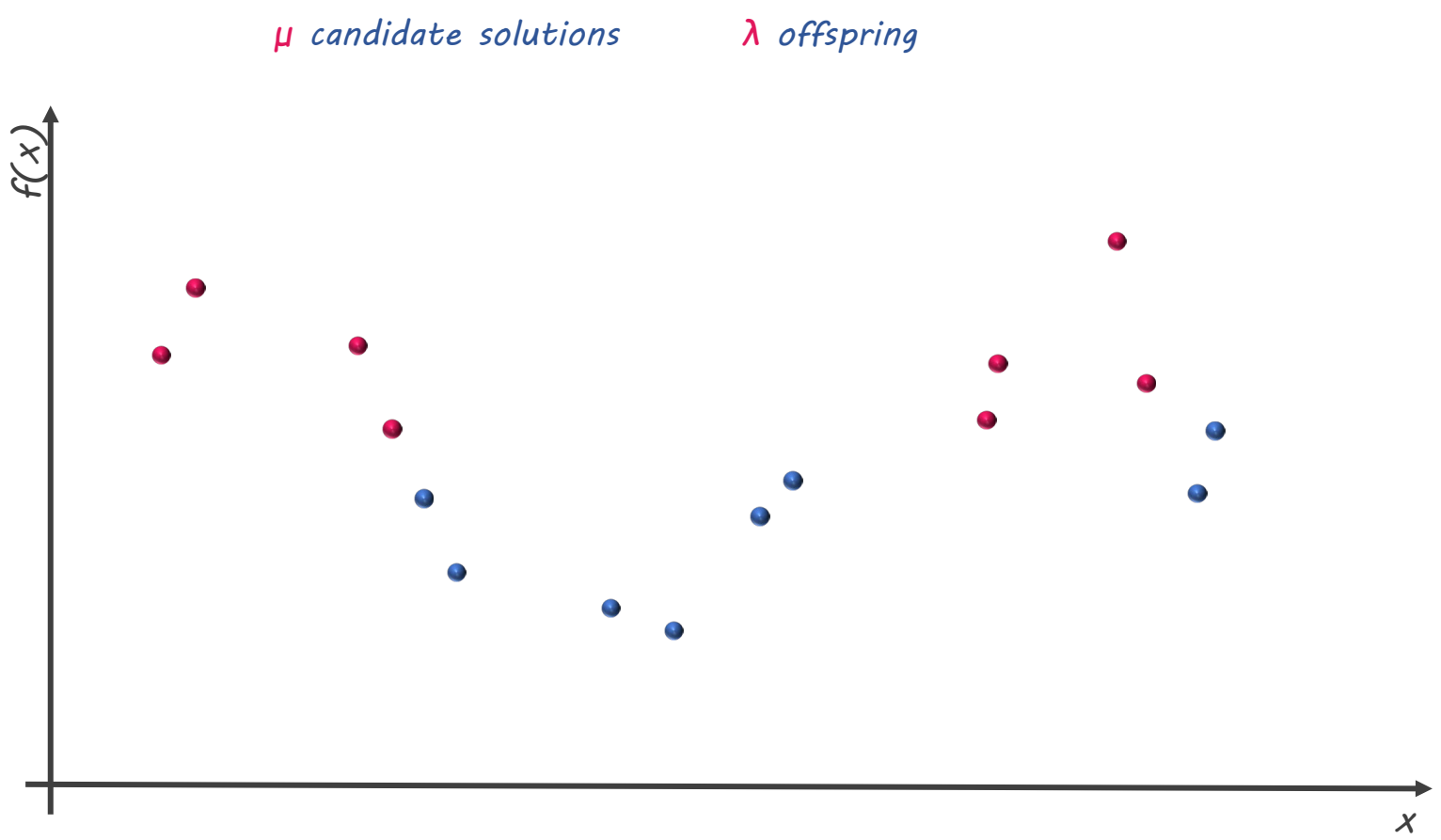
$(\mu + \lambda)$ES



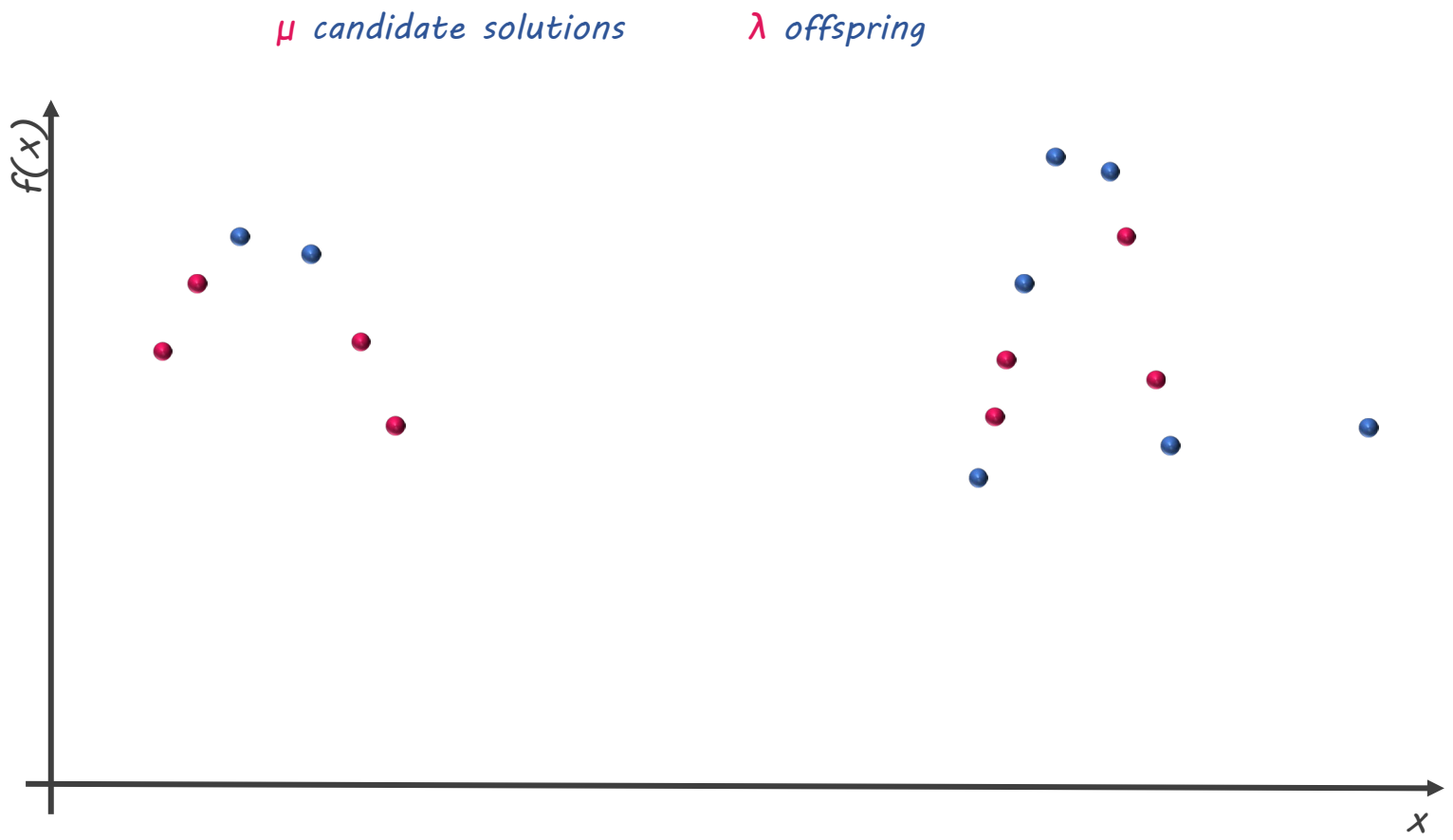
$(\mu + \lambda)$ES



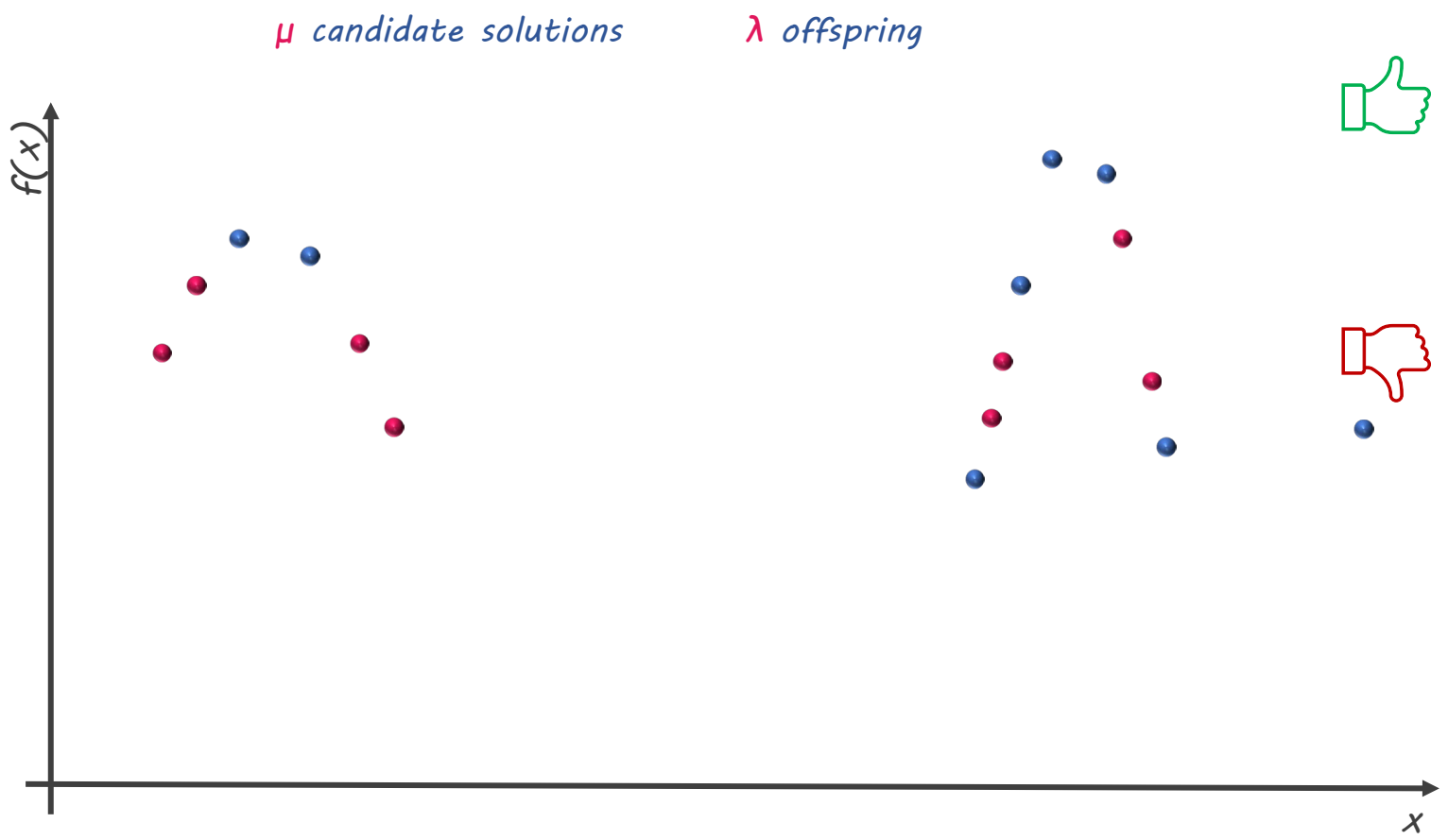
$(\mu + \lambda)$ES



$(\mu + \lambda)$ES



$(\mu + \lambda)$ES



elitism preserves best-so-far solutions

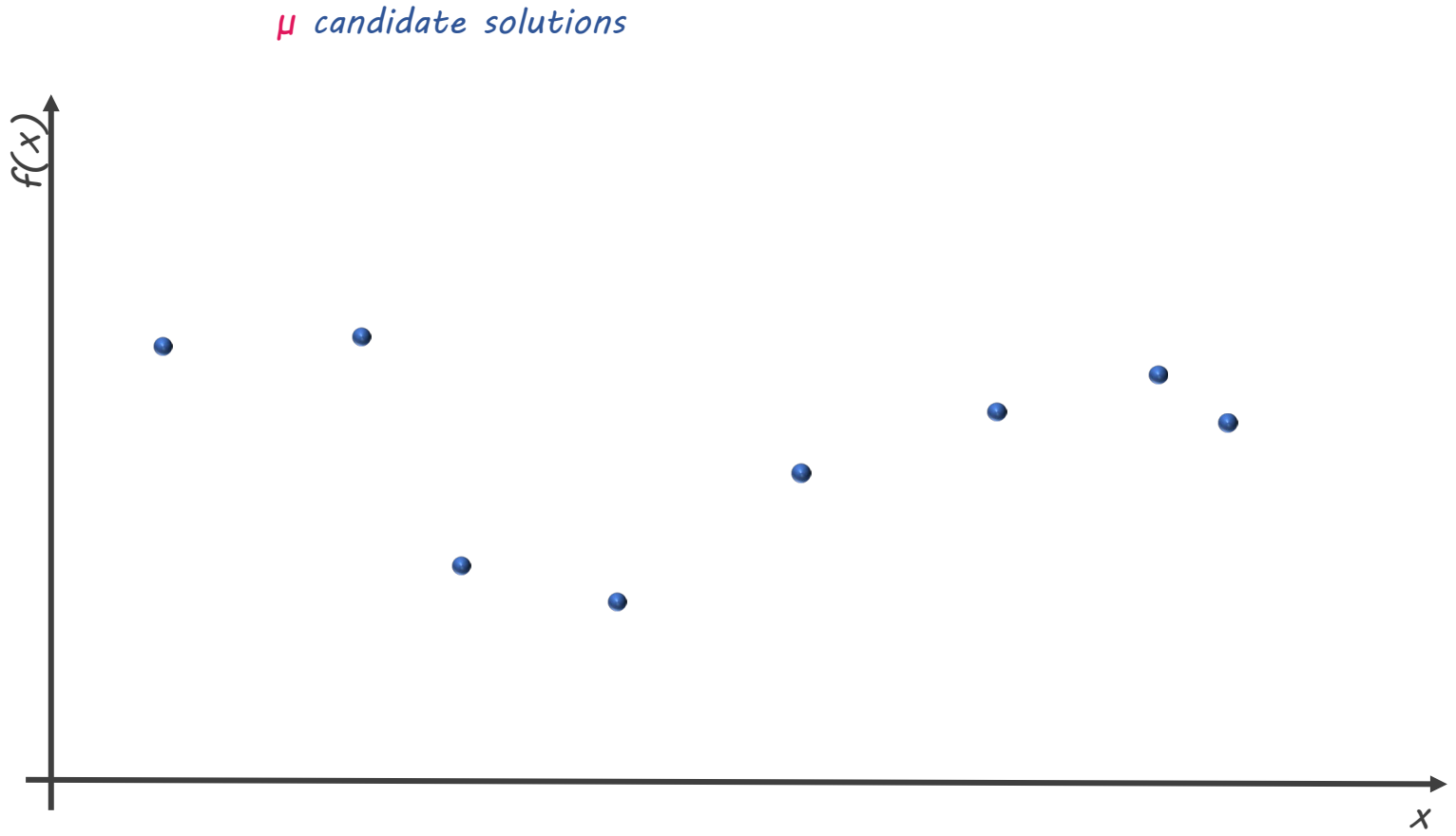
sensitive to μ and μ values

premature convergence

(μ, λ)ES

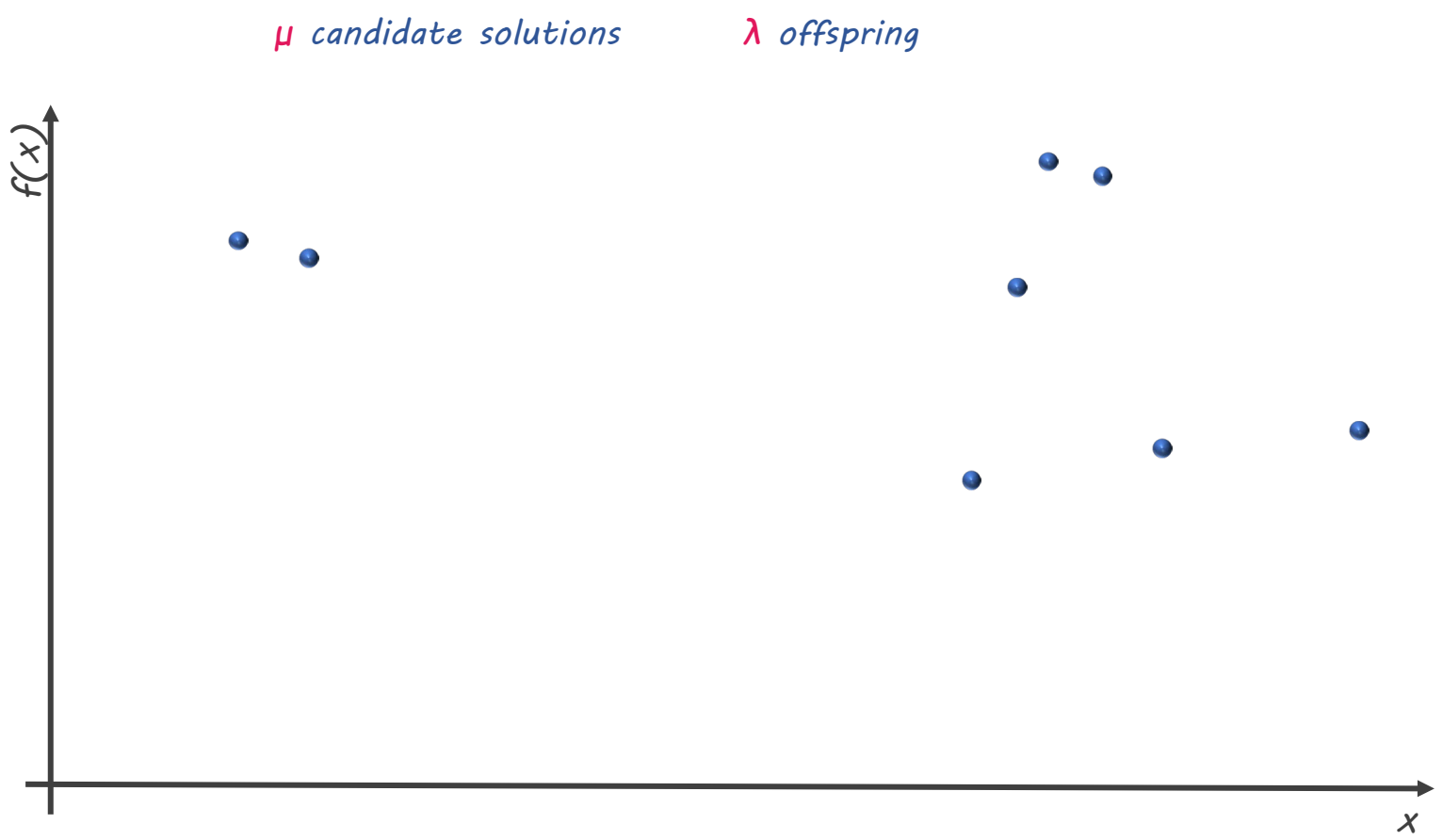
- population size: $\mu + \lambda$
- evolution: the μ best candidate solutions generate λ offspring
- selection: the offspring always replace the parents

(μ, λ)ES

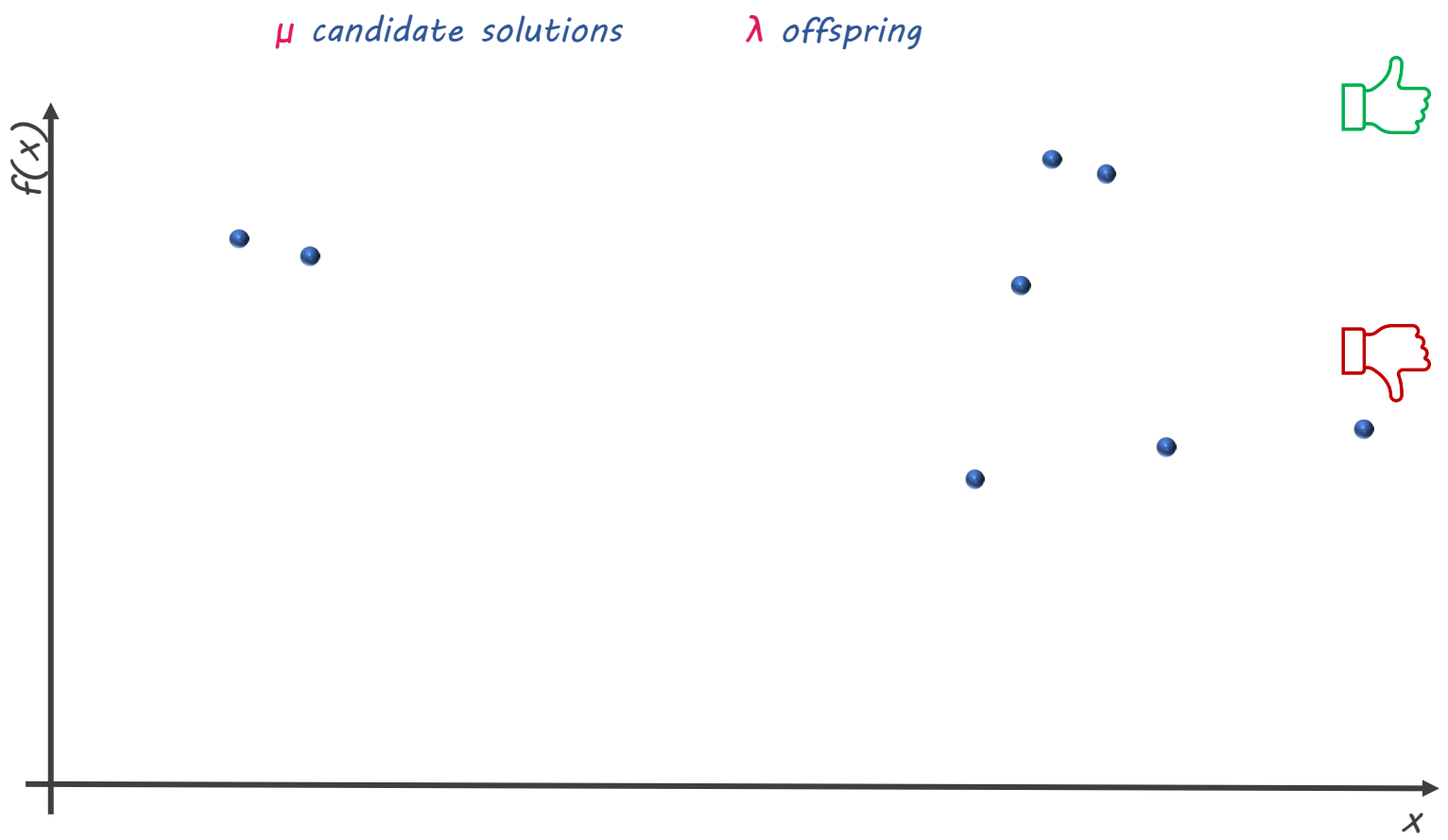


(μ, λ)ES

(μ, λ)ES



(μ, λ)ES



avoiding local optima

useful for dynamic optimization

sensitive to μ and μ values

slow convergence

Evolution Strategies (ES)

Of course, several versions exist, e.g.:

- $(1, \lambda)$
- $(1 + \lambda)$
- $(\mu + 1)$
- $(^{\mu}/_{\rho} +, \lambda)$
- …

faster improvements, but may lead to premature convergence – $\mu > \lambda$

so that multiple parent options (i.e., strategies) exist – $\mu > 1$

surplus of offspring ensures exploration – $\lambda > \mu$

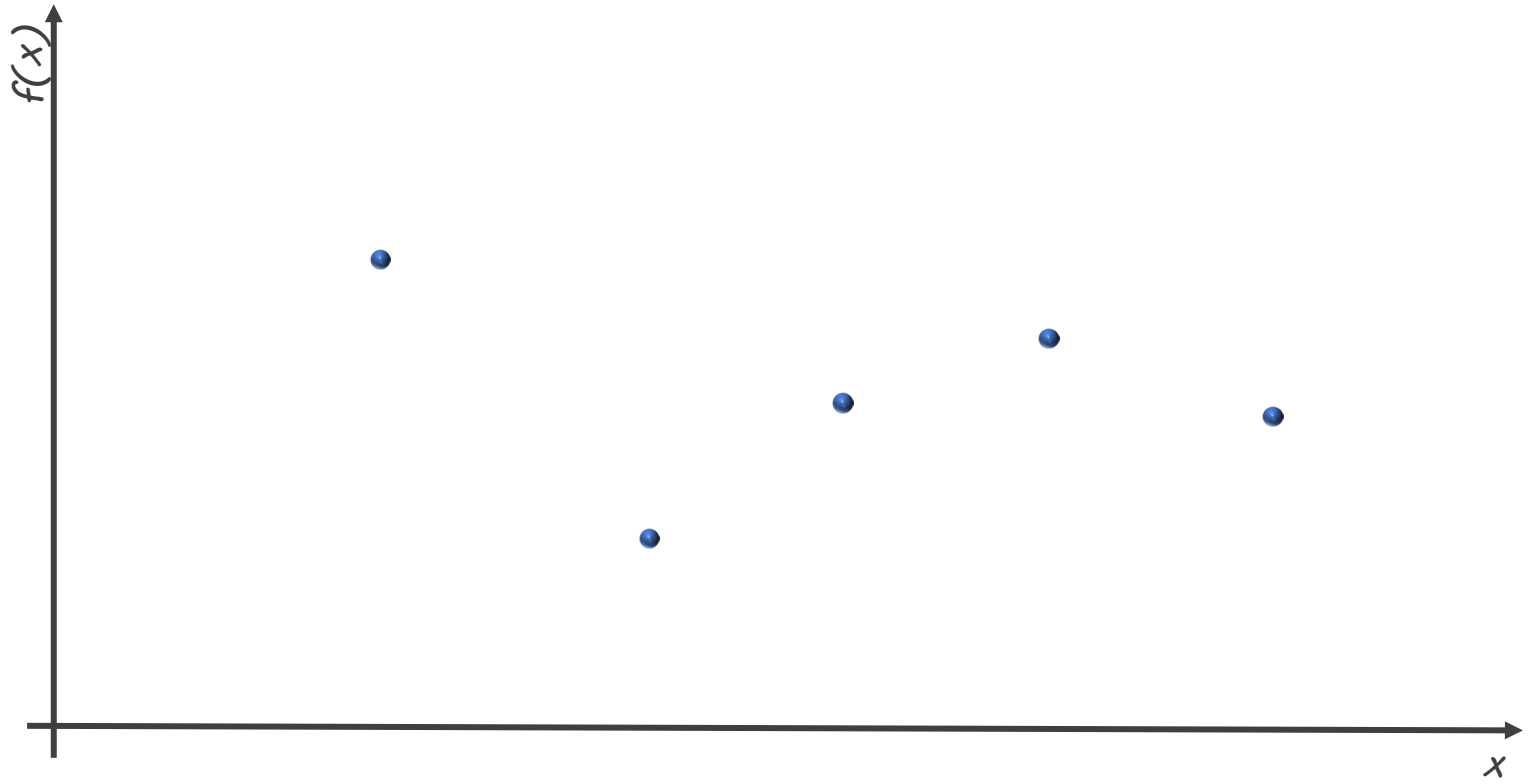
typically, $\lambda > \mu$ – *induces a large selection pressure*

Selection pressure

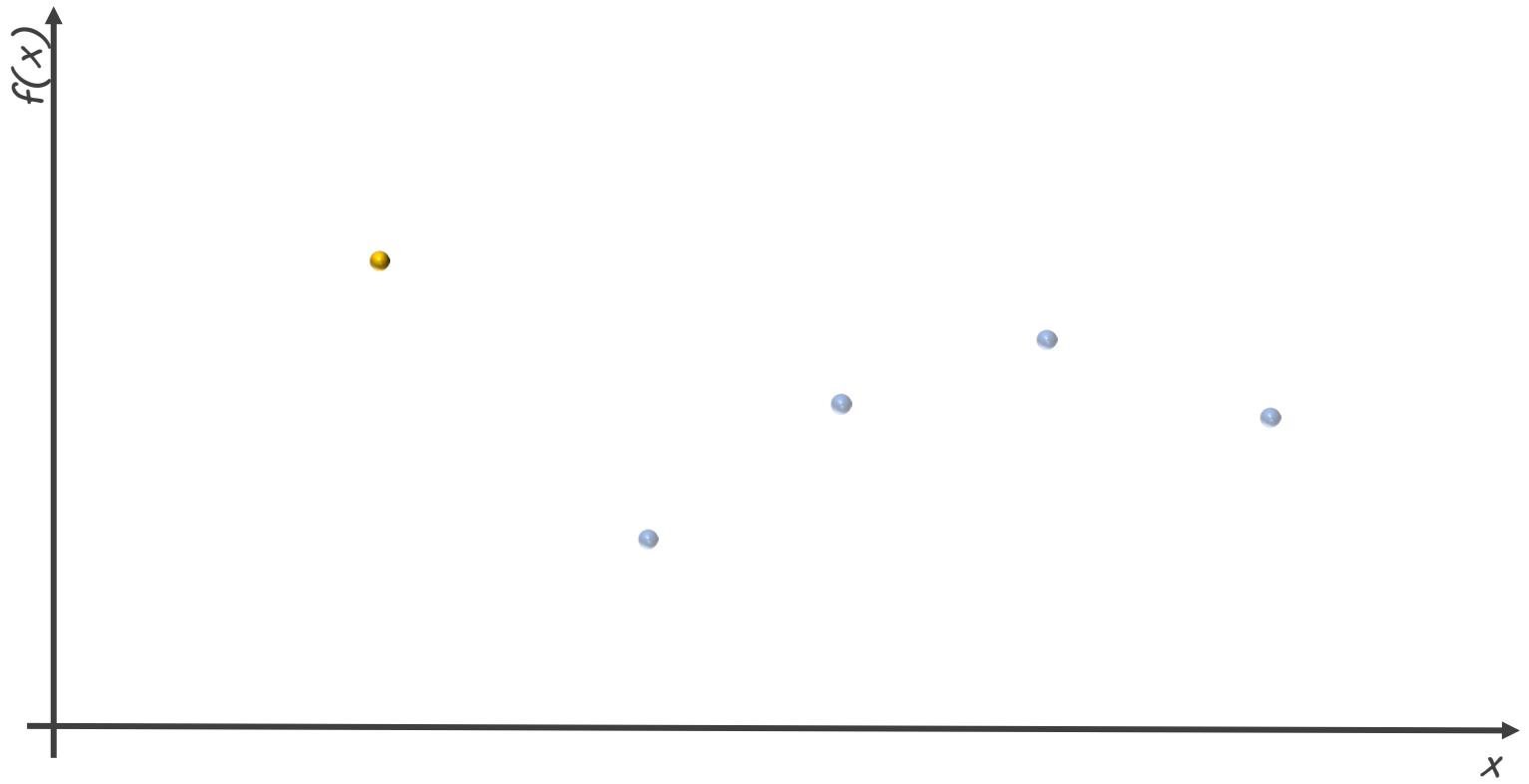
selection pressure: describes how possible is for fitter candidate solutions to be selected

$^{\lambda}/_{\mu} \approx 5 - 7$

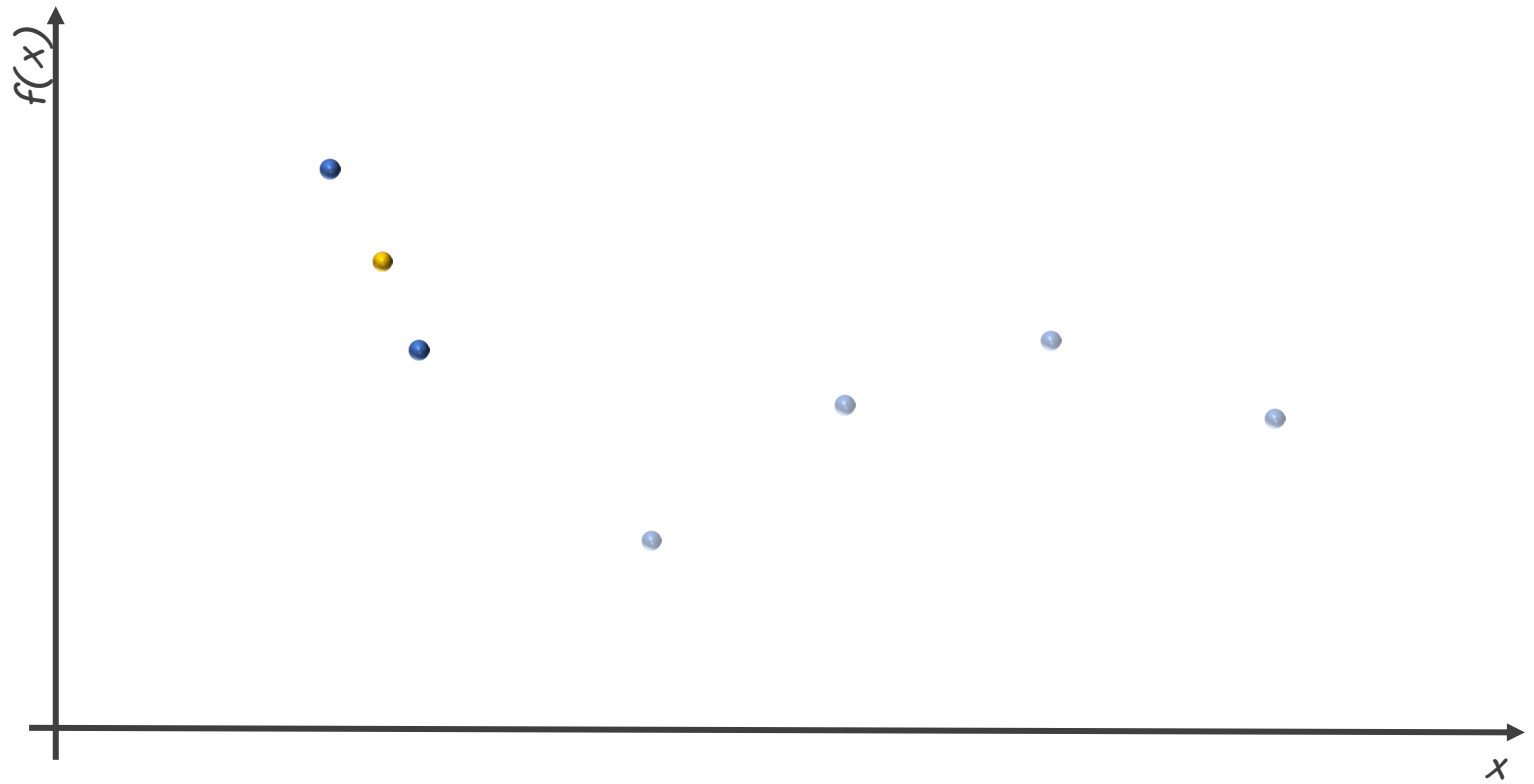
lower μ - too few



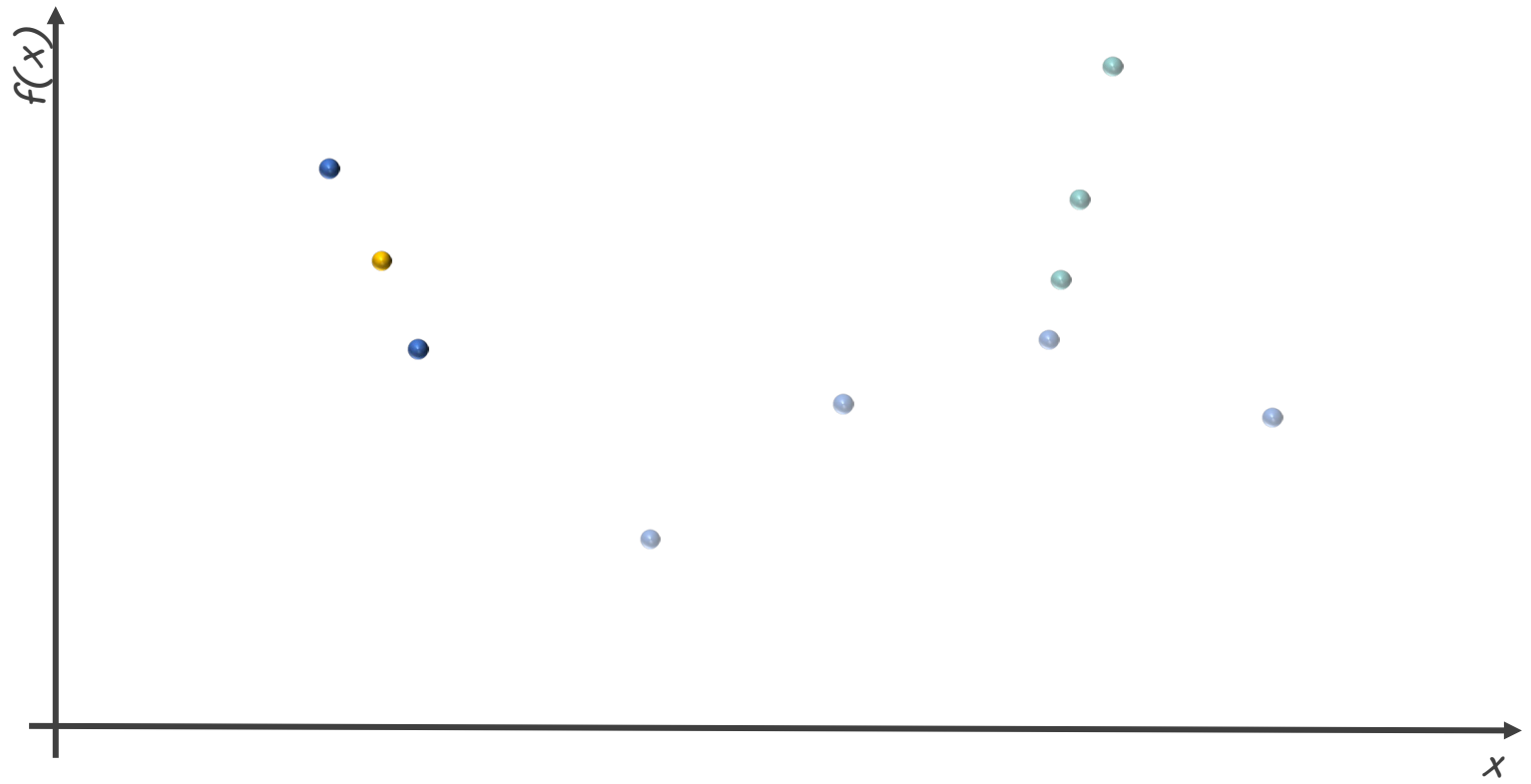
lower μ - too few



lower μ - too few

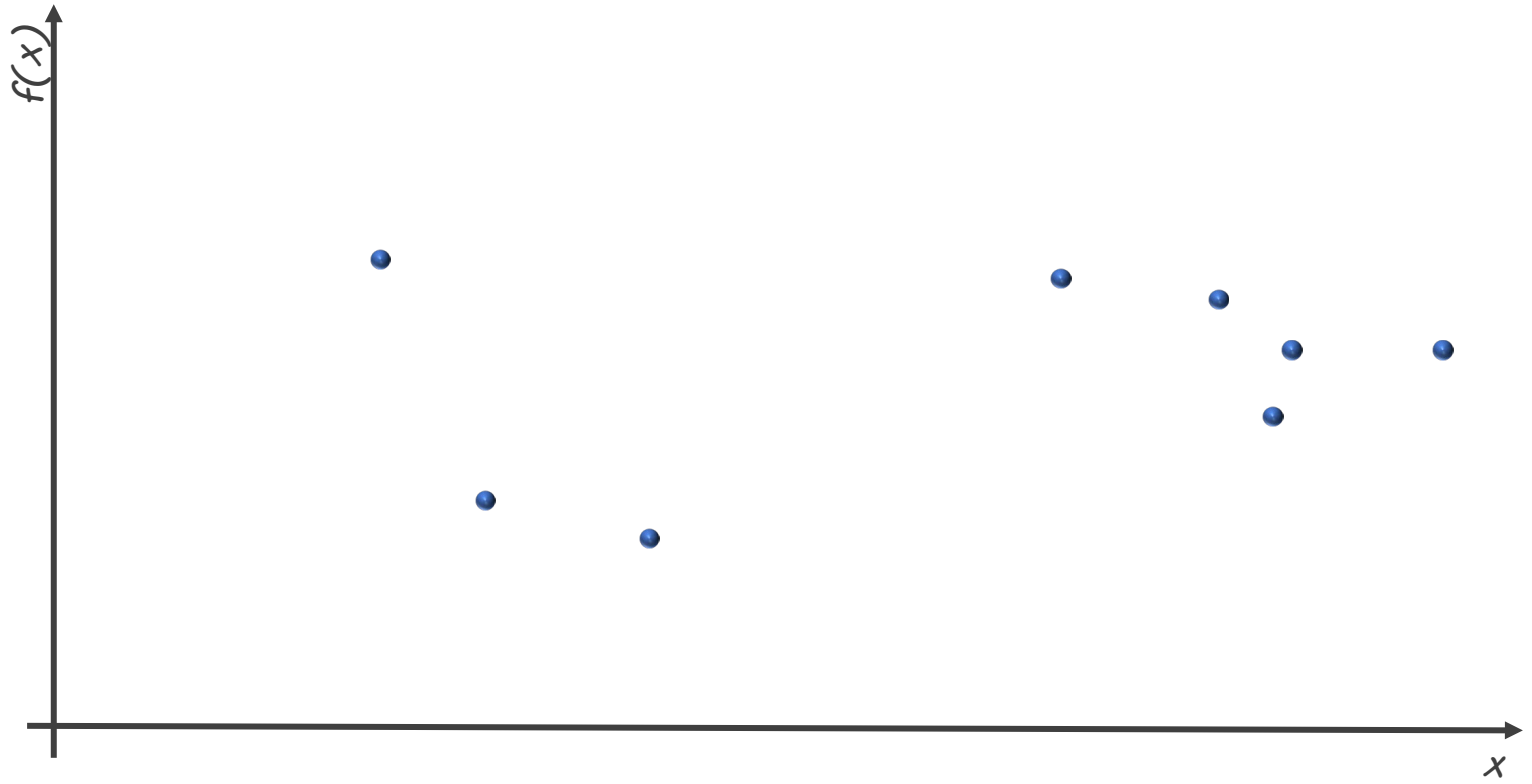


lower μ - too few



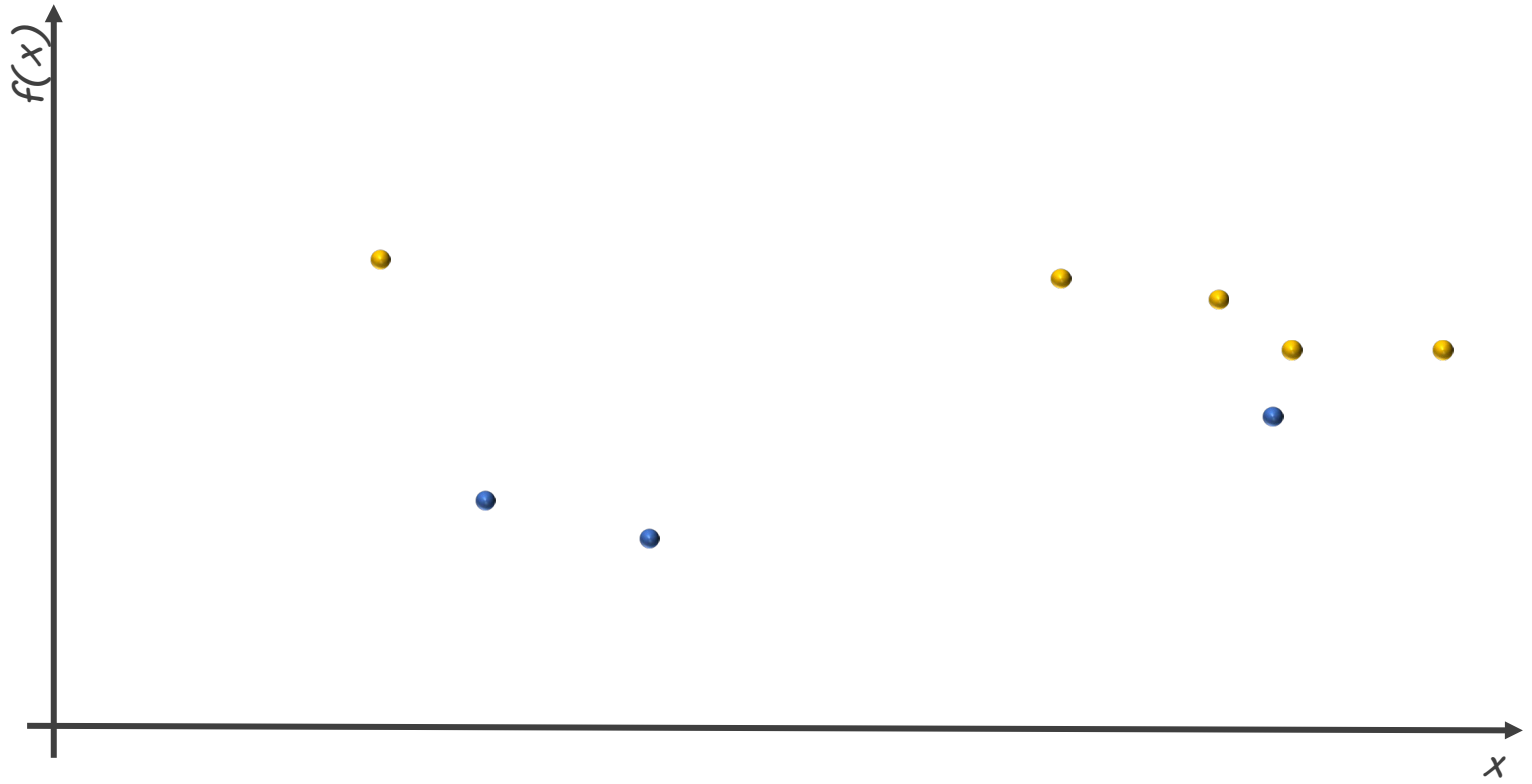
higher μ - too many

lower λ - too few



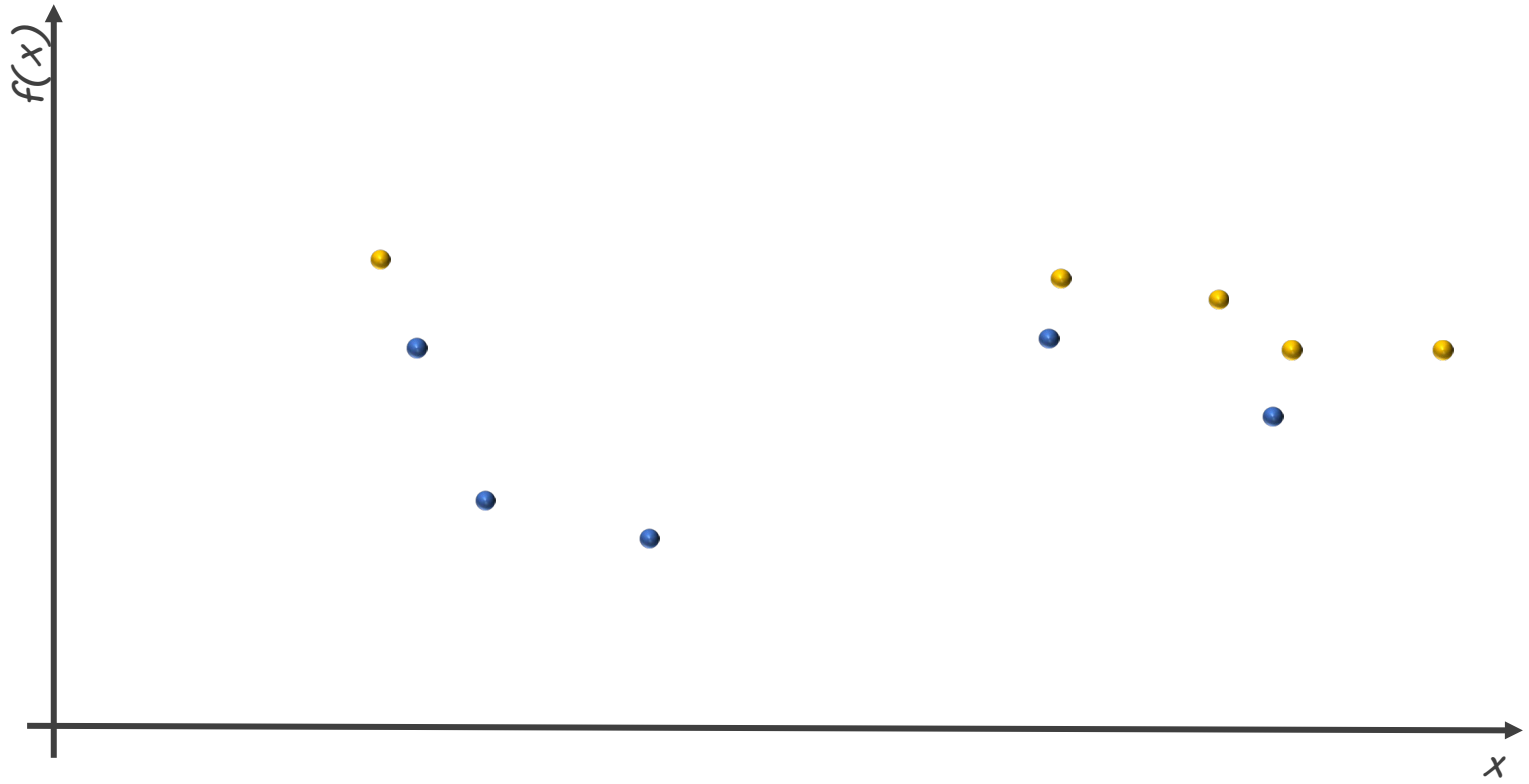
higher μ - too many

lower λ - too few



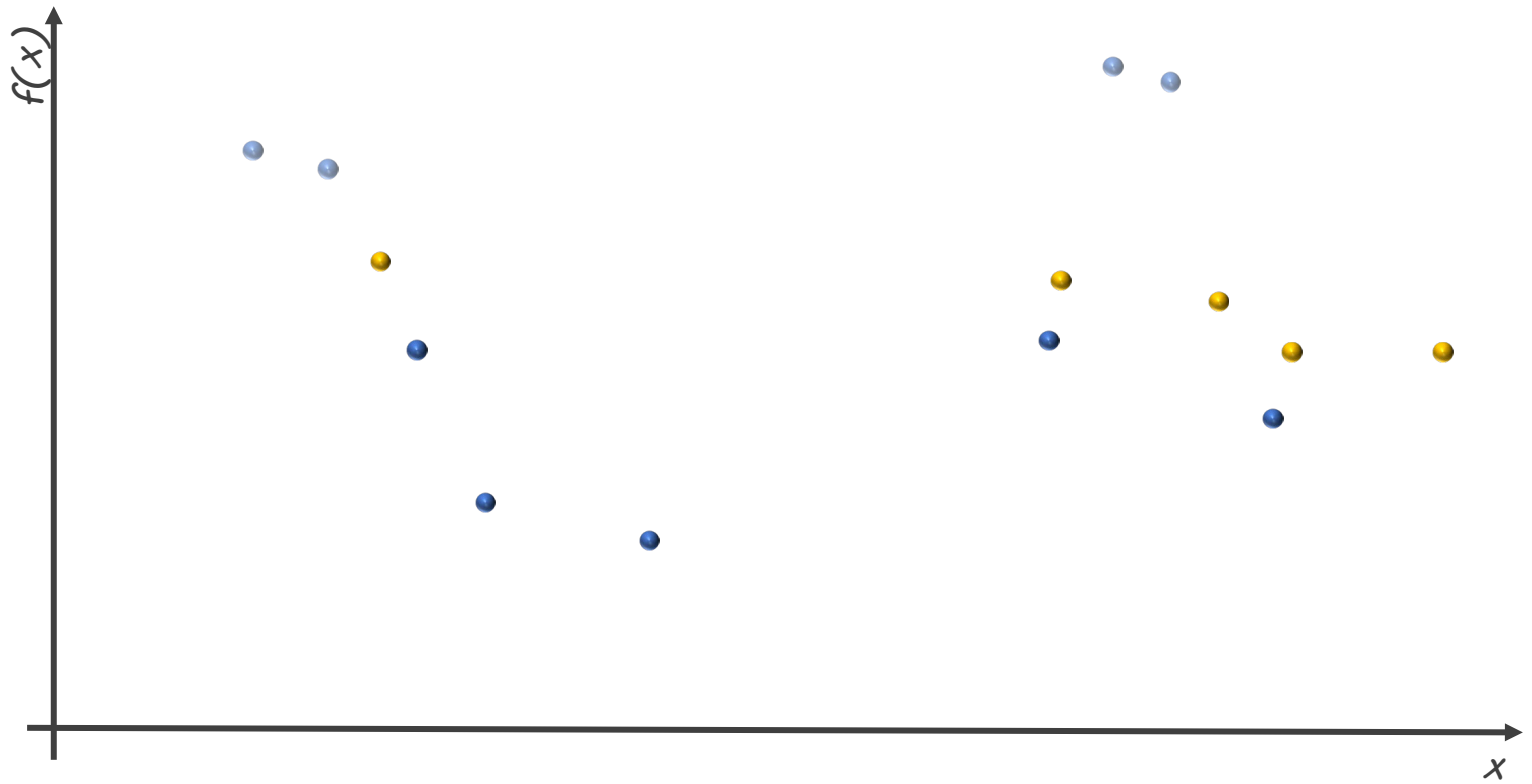
higher μ - too many

lower λ - too few



higher μ - too many

lower λ - too few



Adaptive mutation strength

How do we decide the mutation strength?

…we don't!

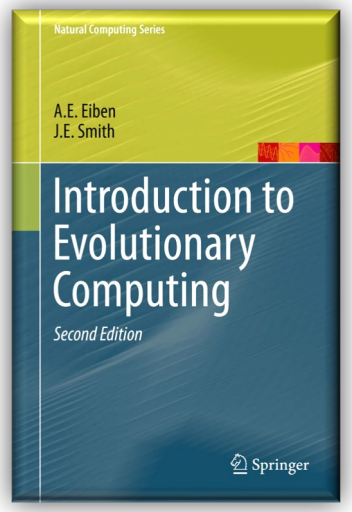
A powerful concept in ES is called self-adaptation.

One way to apply self-adaptation is to use the 1/5 success rule of Rechenberg:

- *more* than $1/5$ offspring are better than the parents: too much exploitation ⇨ increase the mutation strength
- *less* than $1/5$ offspring are better than the parents: too much exploration ⇨ decrease the mutation strength
- *exactly* $1/5$ offspring are better than the parents: ⇨ don't change anything

JÖNKÖPING UNIVERSITY

Lecture material

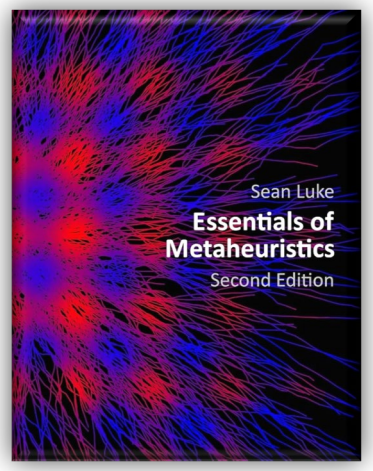


Eiben, A. E., Smith, J. E., Eiben, A. E., & Smith, J. E. (2015). Fitness, Selection, and Population Management. Introduction to Evolutionary Computing. https://doi.org/10.1007/978-3-662-44874-8_5 (special focus on section 5.4 – Selection pressure)

Eiben, A. E., Smith, J. E., Eiben, A. E., & Smith, J. E. (2015). Popular Evolutionary Algorithm Variants. Introduction to Evolutionary Computing. https://doi.org/10.1007/978-3-662-44874-8_6 (special focus on section 6.2 – Evolution Strategies)

Luke, S. (2009). Single-State Methods. Essentials of Metaheuristics, http://cs.gmu.edu/~sean/book/metaheuristics/

Luke, S. (2009). Population Methods. Essentials of Metaheuristics, http://cs.gmu.edu/~sean/book/metaheuristics/




JÖNKÖPING UNIVERSITY