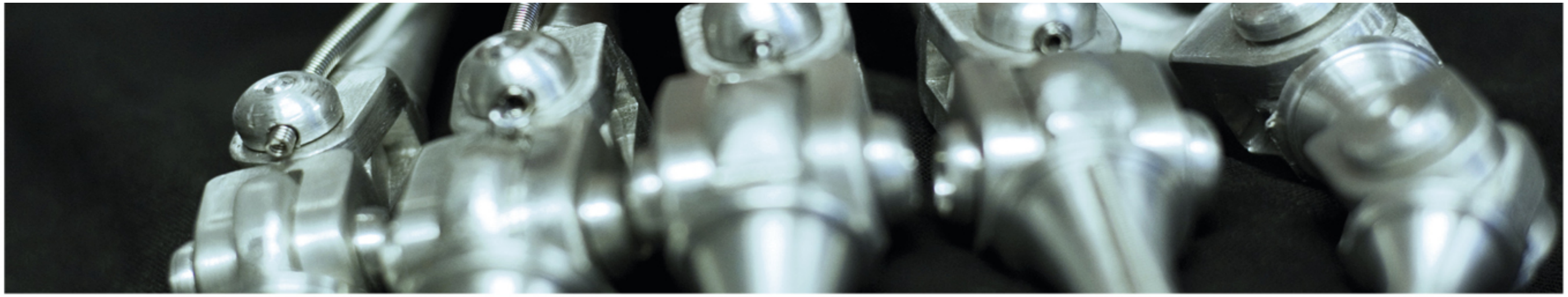


UiO : Department of Informatics
University of Oslo

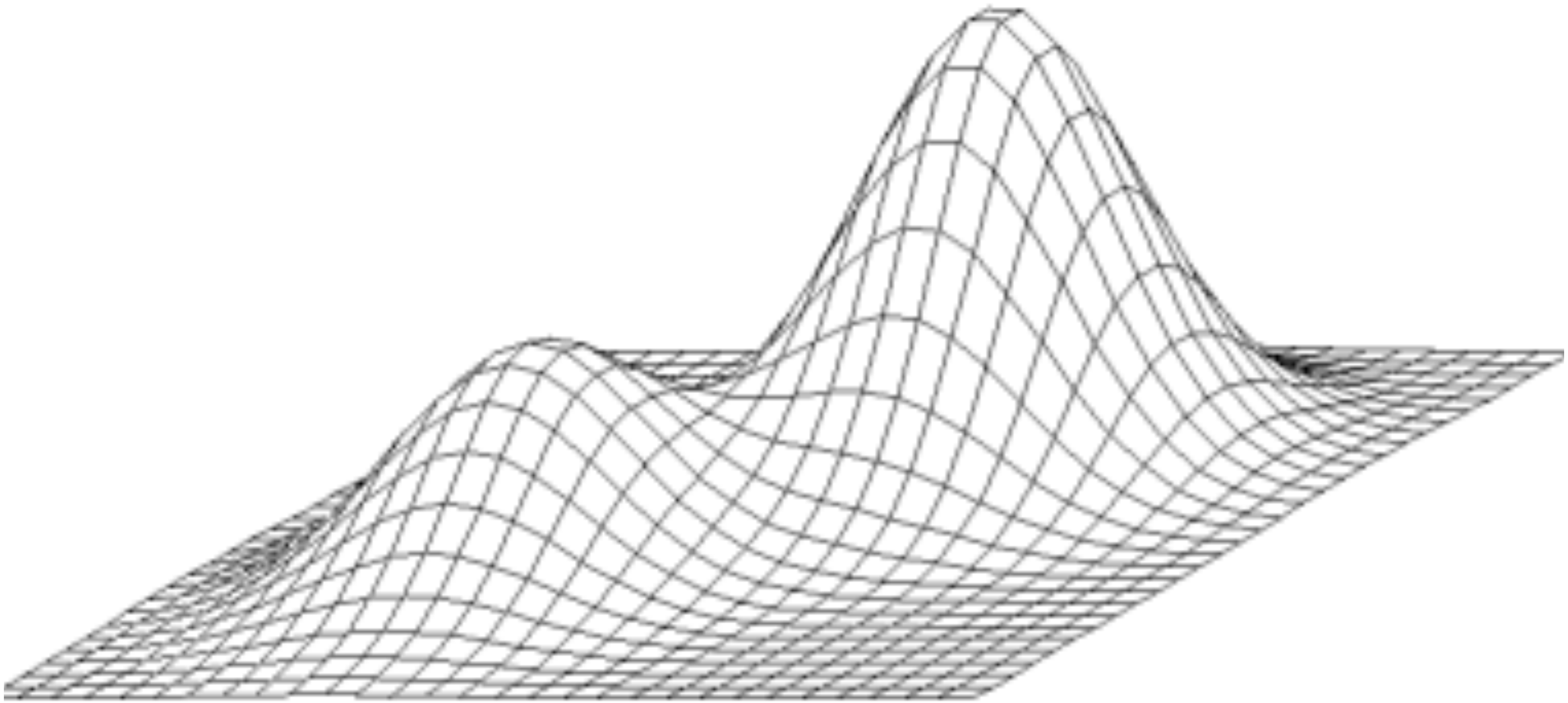
INF3490 - Biologically inspired computing

Lecture 1: Marsland chapter 9.1, 9.4-9.6

Optimization and Search

Jim Tørresen

Optimization and Search



Optimization and Search Methods (selection)

1. Exhaustive search
2. Greedy search and hill climbing
3. Gradient ascent
4. Simulated annealing

Optimization

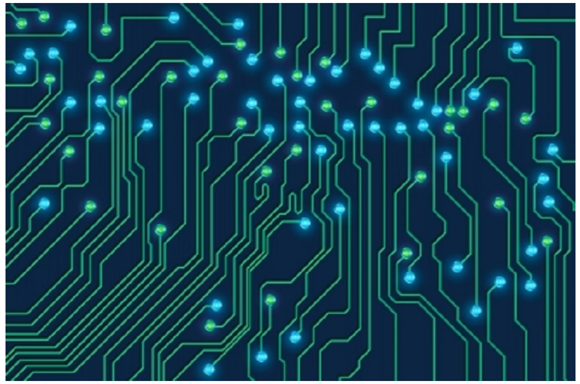
We need

- A numerical representation x for all possible solutions to the problem
- A function $f(x)$ that tells us how good solution x is
- A way of finding
  - $\max_{x} f(x)$ if bigger $f(x)$ is better (benefit)
  - $\min_{x} f(x)$ if smaller $f(x)$ is better (cost)

Optimisation and Search

- **Continous Optimization** is the mathematical discipline which is concerned with finding the maxima and minima of functions, possibly subject to constraints.

- **Discrete Optimization** is the activity of looking thoroughly in order to find an item with specified properties among a collection of items.

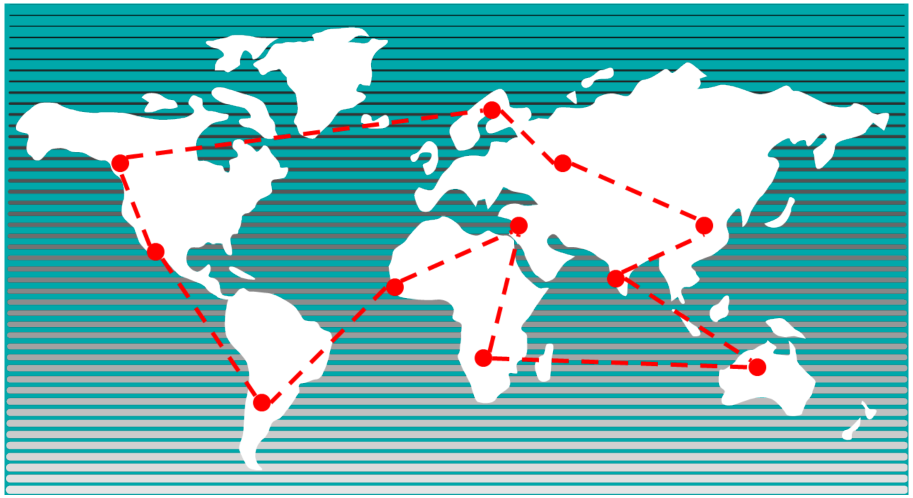


Discrete optimization

- **Chip design**
  - Routing tracks during chip layout design
- **Timetabling**
  - E.g.: Find a course time table with the minimum number of clashes for registered students
- **Travelling salesman problem**
  - Optimization of travel routes and similar logistics problems

Example: Travelling Salesman Problem (TSP)

- Given the coordinates of *n* cities, find the ***shortest closed tour*** which visits each ***once and only once*** (i.e. exactly once).
- Constraint :
  - all cities be visited, once and only once.



1. Exhaustive search

- Test all possible solutions, pick the best
- Guaranteed to find the optimal solution

Exhaustive search

Only works for simple discrete problems, but can be approximated in continuous problems

- Sample the space at regular intervals (grid search)
- Sample the space randomly N times

2. Greedy search

- Pick a solution as the current best
- Compare to all neighboring solutions
  - If no neighbor is better, then terminate
  - Otherwise, replace the current best with the best of the neighbors
  - Repeat

Hill climbing

- Pick a solution as the current best
- Compare to a random neighbor
  - If the neighbor is better, replace the current best
  - Repeat

Continuous optimization

- **Mechanics**
  - Optimized design of mechanical shapes etc.
- **Economics**
  - Portfolio selection, pricing options, risk management etc.

- **Control engineering**
  - Process engineering, robotics etc.

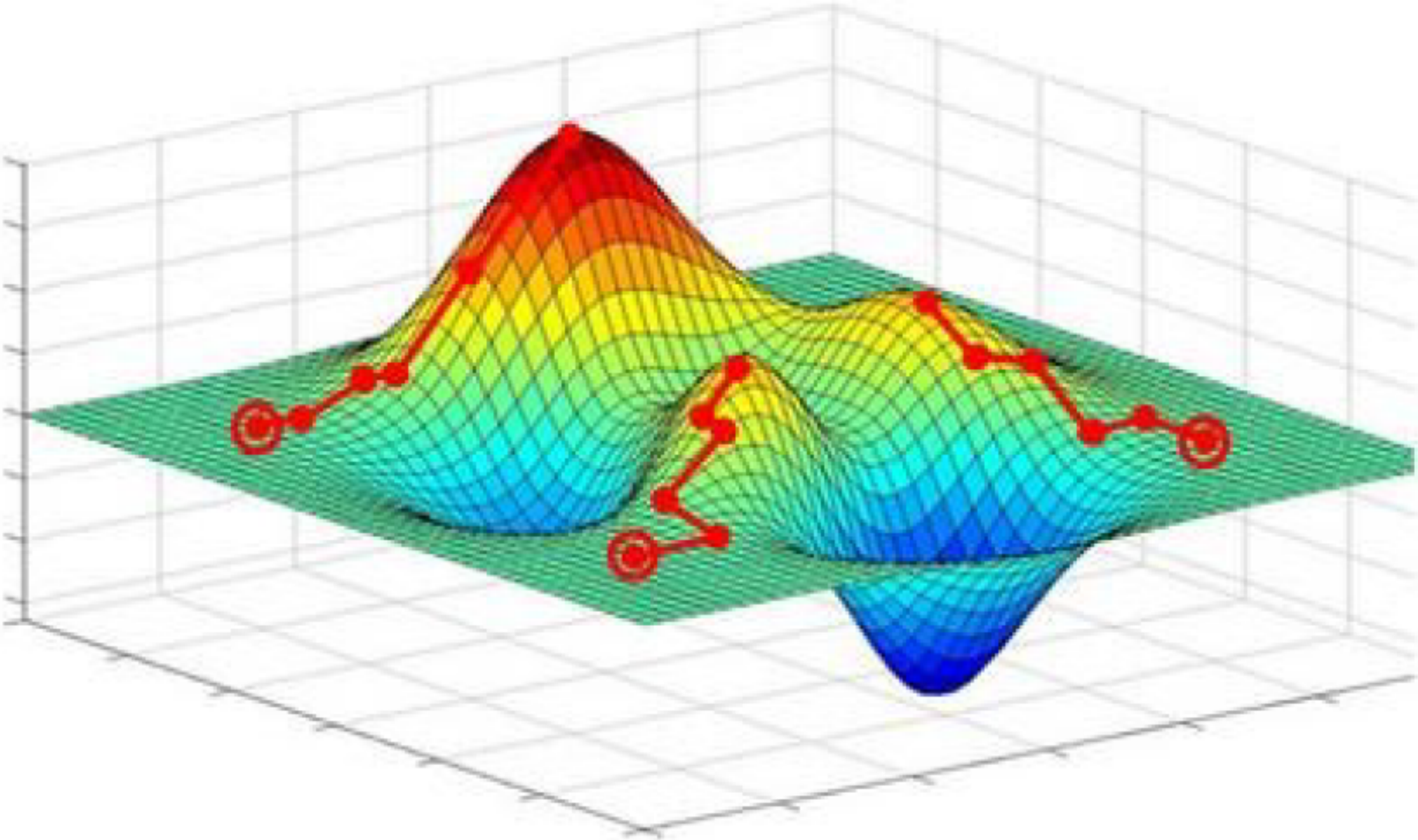
3. Gradient ascent / descent

In continuous optimization we may be able to calculate the gradient of $f(x)$:

$$\nabla f(x) = \begin{bmatrix} \frac{\delta f(x)}{\delta x_0} \\ \frac{\delta f(x)}{\delta x_1} \\ \vdots \\ \frac{\delta f(x)}{\delta x_n} \end{bmatrix}$$

The gradient tells us in which direction $f(x)$ increases the most

3. Gradient ascent / descent



Gradient ascent / descent

Starting from $x^{(0)}$, we can iteratively find higher $f\left(x^{(k+1)}\right)$ by adding a value proportional to the gradient to $x^{(k)}$:

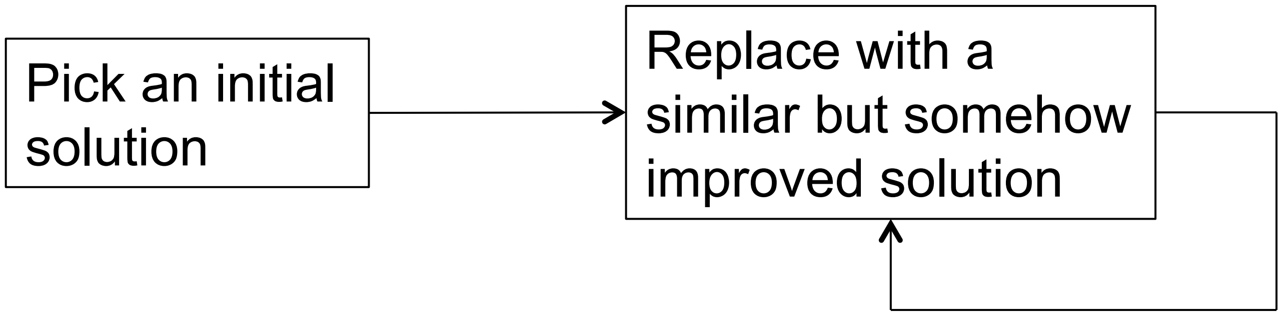
$$x^{(k+1)} = x^{(k)} + \gamma \nabla f\left(x^{(k)}\right)$$

Local optima

Algorithms like greedy search, hill climbing and gradient ascent/descent can only find local optima:

- They will only move through a strictly improving chain of neighbors
- Once they find a solution with no better neighbors they stop

Exploitation



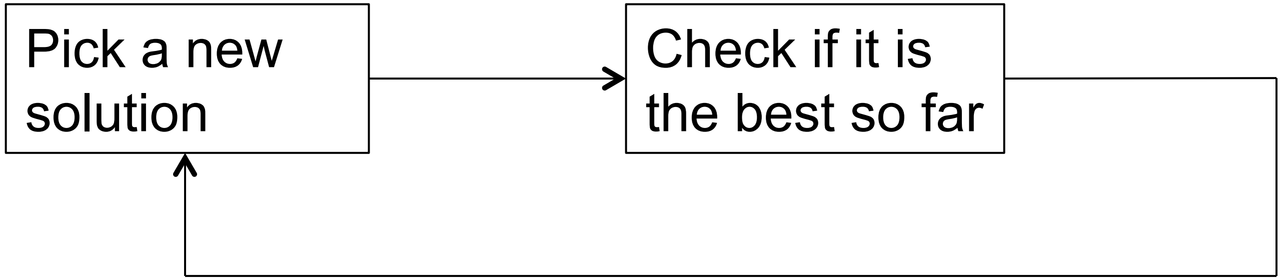
Global optimization

- Most of the time, we must expect the problem to have many local optima
- Ideally, we want to find the best local optima: the global optimum

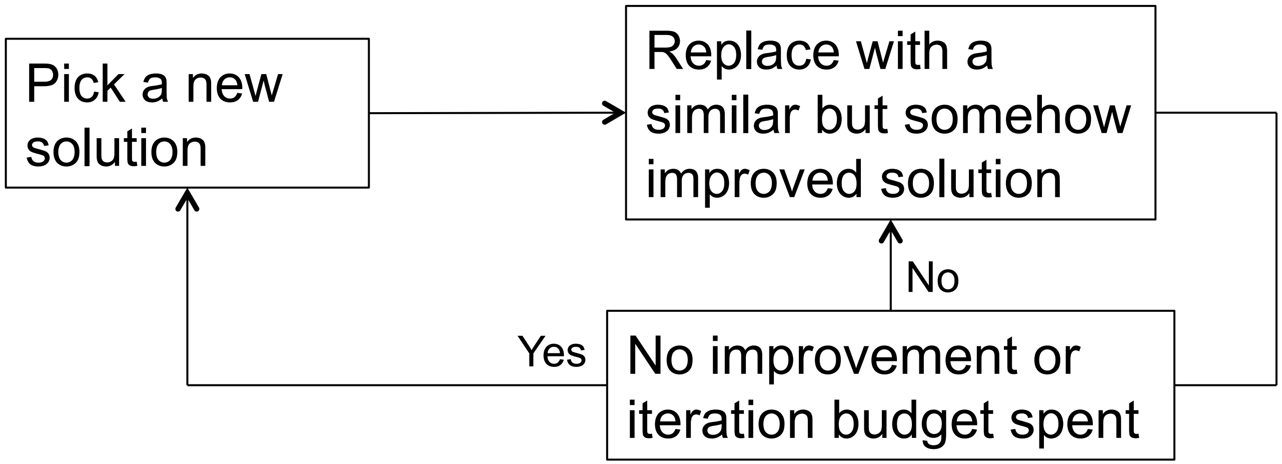
Exploitation and Exploration

- Search methods should combine:
  - Trying completely new solutions (like in exhaustive search) => **Exploration**
  - Trying to improve the current best solution by local search => **Exploitation**

Exhaustive search – pure exploration



Mixed solution



Works better, but only if there are few local optima

Going the wrong way

What if we modified the hill climber to sometimes choose worse solutions?

- Goal: avoid getting stuck in a local optimum
- Always keep the new solution if it is better
- However, if it is worse, we'd still want to keep it sometimes, i.e. with some probability

Annealing

A thermal process for obtaining low energy states of a solid in a heat bath:

- Increase the temperature of the heat bath to a the point at which the solid melts
- Decrease the temperature slowly
- If done slowly enough, the particles arrange themselves in the minimum energy state

4. Simulated annealing

- Set an initial temperature T
- Pick an initial solution
- Repeat:
  - Pick a solution neighboring the current solution
  - If the new one is better, keep it
  - Otherwise, keep the new one with a probability
    $$P(\Delta f, T) = e^{-\Delta f / T}$$
  - Decrease T