Why are there so many different optimization algorithms?

This week's video lectures discuss several optimization strategies:

- Exhaustive search
- Greedy search
- Hill climbing
- Gradient descent/ascent
- Simulated annealing

(+ Evolutionary algorithms (next week))

Why are there so many different methods?
Does it matter in practice which one I use?

Local optima

The video lectures point out that some algorithms can only find *local optima,* for example greedy search and hill climbing.

- Why do many problems lead to the existence of local optima?

- How do I know if an algorithm has found only a local optimum and not a globally optimal solution?

- What can I expect from a local optimum? Is it ok to use it, or should I always strive for a global optimum?

Finding the top of a mountain

All optimization can be visualized as finding the top of a mountain (or equivalently: the bottom of a valley).

How can we visualize the different optimization strategies in this context?

- Exhaustive search
- Greedy search
- Hill climbing
- Gradient descent/ascent
- Simulated annealing

Simulated annealing

- What is the temperature T in simulated annealing? Why is it called temperature?

- In simulated annealing the temperature should be decreased during the optimization. Why?

- How do I select start temperature?

- How fast should I reduce the temperature?