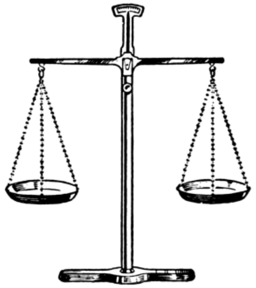
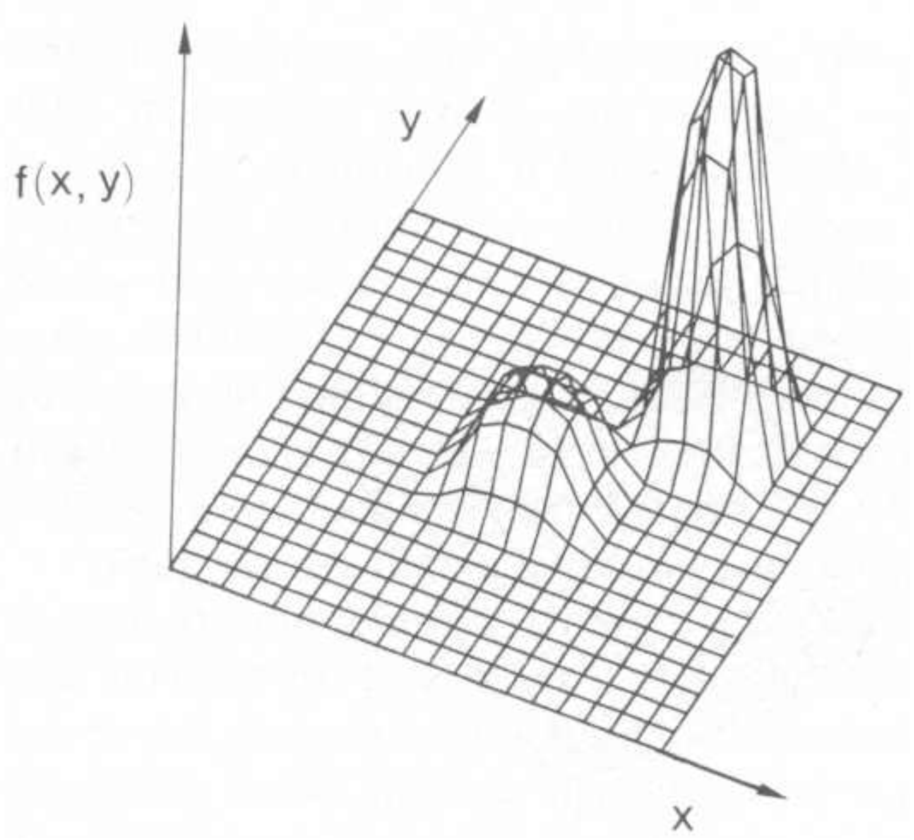


Metaheuristics

Local Optimization vs Global Optimization

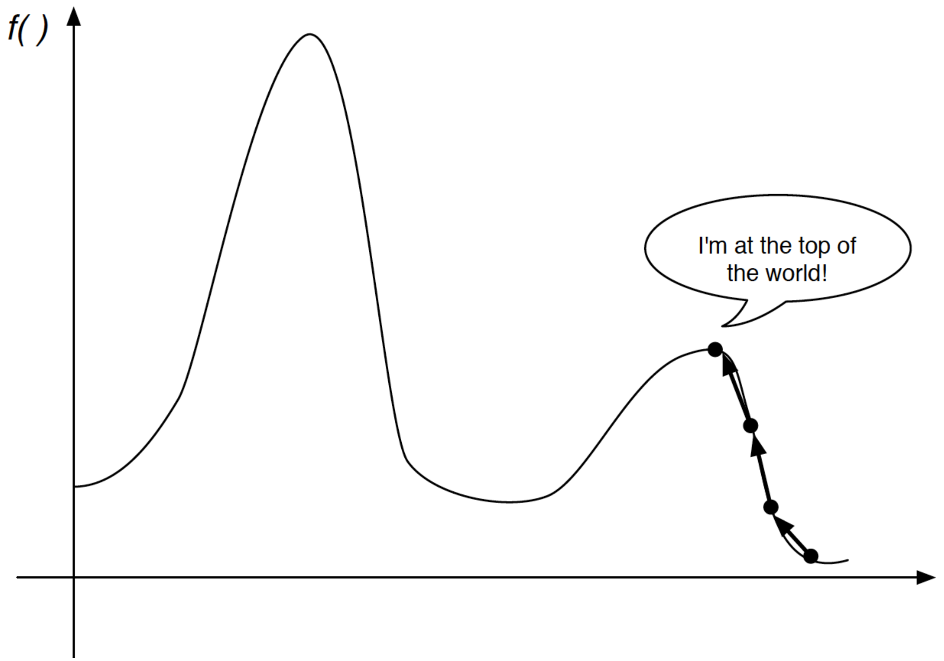
Local optimization:
start from one point and climb the hill
up to the top

No guarantee to reach a global
optimum !!!



Local Optimization vs Global Optimization

Let us consider a classical optimization method as the gradient algorithm.



- How do you escape local optima?

Local Optimization vs Global Optimization

With the tools we have so far, there's really only one way: change the **step size** to a sufficiently large value that the algorithm potentially overshoots not only the top of its hill but actually lands on the next hill.

Alternatively, we could put the Gradient algorithm in a big loop: each time we start with a random starting point, and end when we've reached a local optimum. We keep trying over and over again, and eventually return the best solution discovered.

To determine what the "best solution discovered" is, we need to be able to compute f(x) so we can compare results.

Assuming we have that, we can now construct a global optimization algorithm.

Local Optimization vs Global Optimization

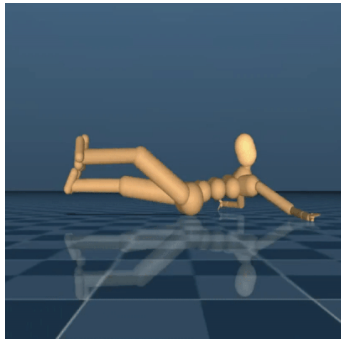
A global optimization algorithm is guaranteed to find the global optimum if it runs long enough.

And at the limit, at some point Gradient Ascent with Restarts will discover the optimum because, at the very least, some restart will randomly land right on the optimum, just like random search.

More realistically, a random restart will eventually start on the globally optimal hill, allowing gradient ascent to climb to the optimum.

Local Optimization vs Global Optimization

- Many classic optimization techniques, such as **Gradient Ascent** make strong assumptions about the nature of f : for example, that we also know its first derivative f 0.



- **Metaheuristics** make far weaker assumptions, and sometimes make none at all. This means that metaheuristics are very general, but also means that they're often best thought of as last-ditch methods, used when no other known technique works.
- That's the case for an enormous, important, and growing collection of problems.

Metaheuristics

To optimize a candidate solution in this scenario, you need to be able to do four things:

- ✓ Provide one or more **initial candidate solutions**. This is known as the initialization procedure.
- ✓ Assess the **Quality** of a candidate solution. This is known as the assessment procedure.
- ✓ Make a Copy of a candidate solution.
- ✓ Tweak a candidate solution, which produces a randomly slightly different candidate solution. This, plus the Copy operation, are collectively known as the **modification** procedure.

To this the algorithm will typically provide a **selection** procedure that decides which candidate solutions to retain and which to reject as it wanders through the space of possible solutions to the problem.

Metaheuristics

- **Metaheuristics** is a term often used to describe a subfield of stochastic optimization.
- **Stochastic optimization** is the general class of algorithms and techniques which employ some degree of randomness to find optimal (or as optimal as possible) solutions to hard problems.
- Metaheuristics are the most general of these kinds of algorithms, and are applied to a very wide range of problems.
- Metaheuristics algorithms are used to find answers to complex problems when you have very little to help you: you don't know beforehand what the optimal solution looks like, you don't know how to go about finding it in a principled way, you have very little heuristic information to go on, and brute-force search is out of the question because the space is too large.

- But if you're given a candidate solution to your problem, you can test it and assess how good it is. That is, you know a good one when you see it.