Monday, March 28 – Day 25

Topic 10 - Introduction to Metaheuristics

So far, we've mostly focused on ways to find optimal solutions.

These techniques are slow, and difficult, aren't always applicable.

Ex: TSP, brute force takes $O(n!)$ time
dynamic programming - $O(n^2 \cdot 2^n)$ time

Metaheuristics:
- General problem solving paradigms that can easily adapted to many problems
- Look for good solutions, and rarely actually find an optimal one.
- Pretty fast

Similar set up:
* Search space made of candidates/solutions
* Every solution has some score / fitness / quality

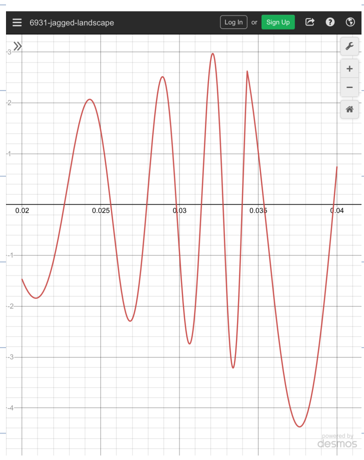
* Goal: Find a candidate with a good score
(assume always maximizing unless specified otherwise)

Many of our problems will be discrete (finite search space), but some will be continuous.

<u>Ex</u>: Find the maximum value of

$$f(x) = \frac{1}{\cos(10(x+1)^2)} \cdot \sin\left(\min\left((x+1)^{100}, \frac{1}{x}\right)\right)$$

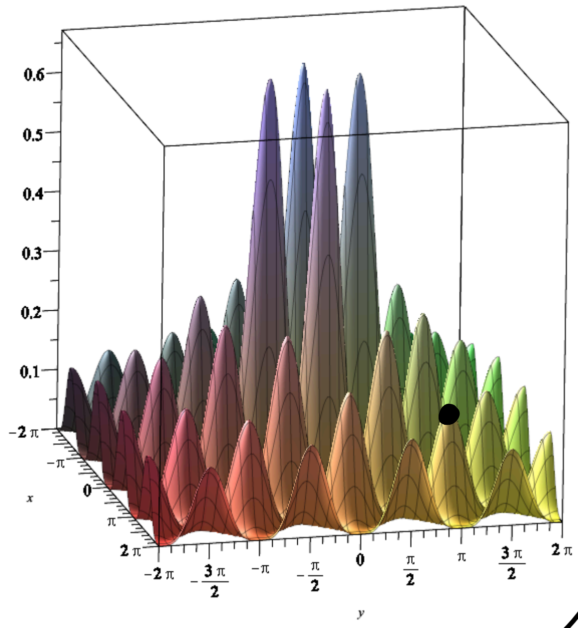
on the interval $0.02 \leq x \leq 0.04$.



Also work on functions that are defined <u>implicity</u>.

Search space: set of real #s from 0.02 to 0.04.

Most of our search spaces will not be 1 dimensional.



$$\frac{\sin^2(x-y)\sin^2(x+y)}{\sqrt{x^2+y^2}}$$

Goal: find the top of the tallest hill

"landscape pictures"

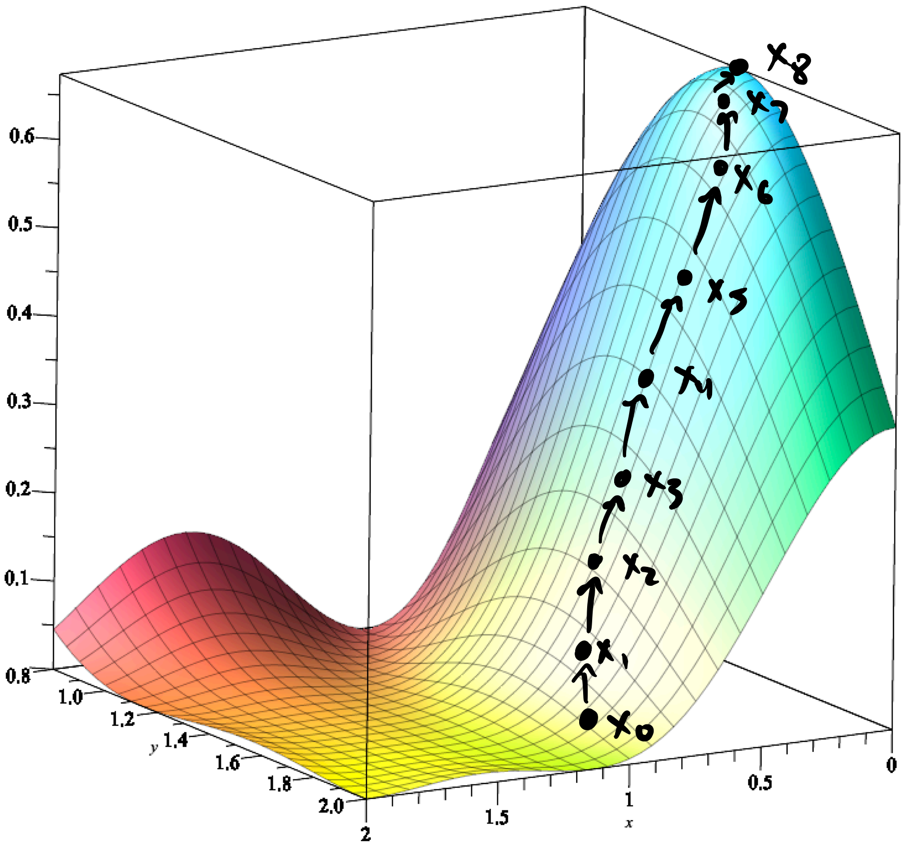
→ without getting stuck on the wrong hilltop

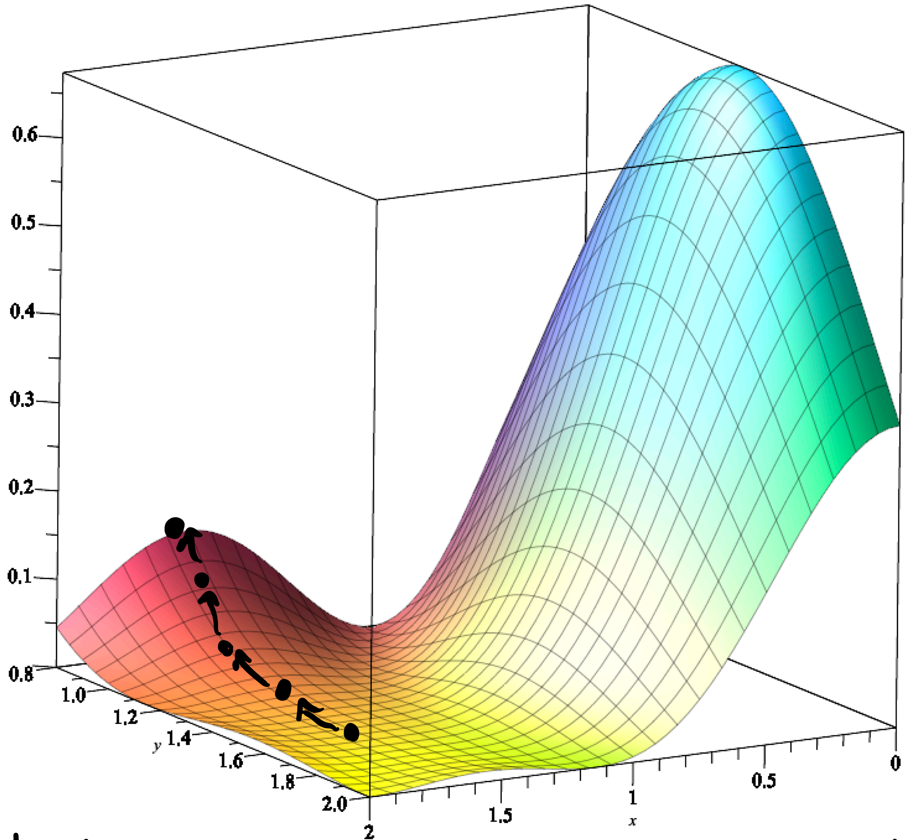
Random: ≈ 21
Random greedy: ≈ 7.12
HC ≈ 6.71
SA ≈ 6.294

Gradient Ascent (or Descent)

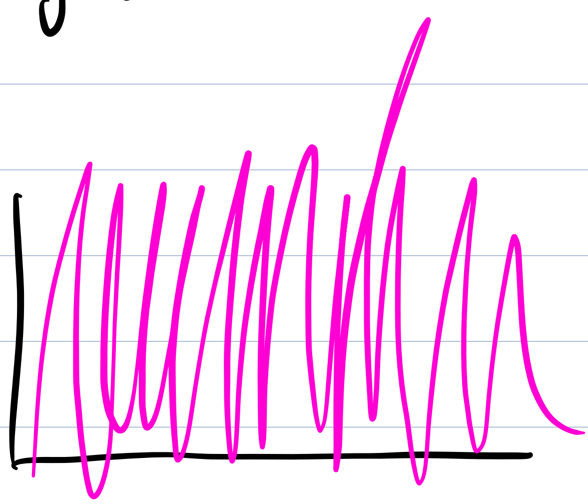
* Optimization method you learn in some classes
* If your function $f(x,y)$ is differentiable, you can compute the gradient at any point. Gradient is a vector that points in the direction of steepest ascent.

(1) Start at a point
(2) compute the gradient at that point
(3) move a little bit in that direction
(4) repeat





Will lead to a <u>local</u> max, but not a global max.



<u>Might overshoot.</u>

How could you simulate Gradient Ascent with a ~~discrete~~ continuous search space.

[pretend you're standing in the mountains]

* start at a random point
* look around yourself in a small radius
  (plug in some points near the current point, and see which is highest)
* go there and repeat

Discrete, TSP as an example
* standing at some possible solution (some tour)
* look at other tours "near" the current tour
* move to the best one and repeat

- need a definition of "nearby" / "small radius"

5 cities, 1, 2, 3, 4, 5

At the solution:

$3 \to 5 \to 2 \to 1 \to 4 \to 3$

What tours are "close" to this one?

One possibility:

nearby = switch two consecutive cities

Others:

nearby = switch any two cities
nearby = take any consecutive block
of cities and reverse them