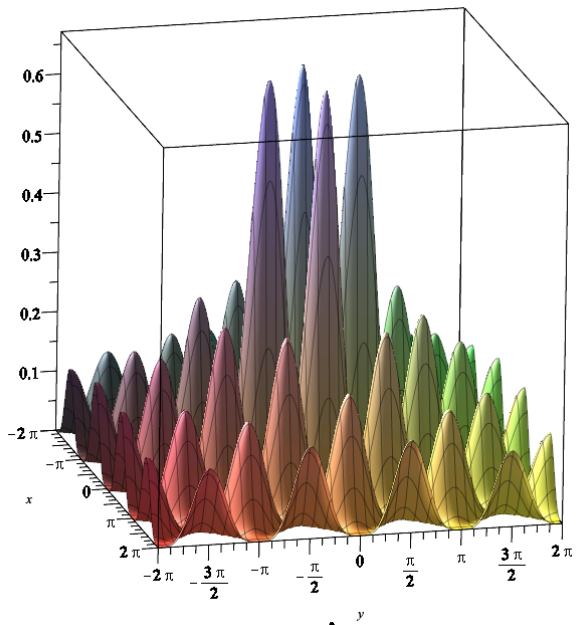
Mon, Mar. 25, 2024
Scientific Computing

Announcements
→ HW 3 feedback posted
→ HW 4 assigned later today
→ No O.H. Thursday
No class Friday
No lecture the next Mon
(or OH)

Topic 9 - Introduction to Metaheuristics (continued)



Landscape Pictures

Search Space Constraints

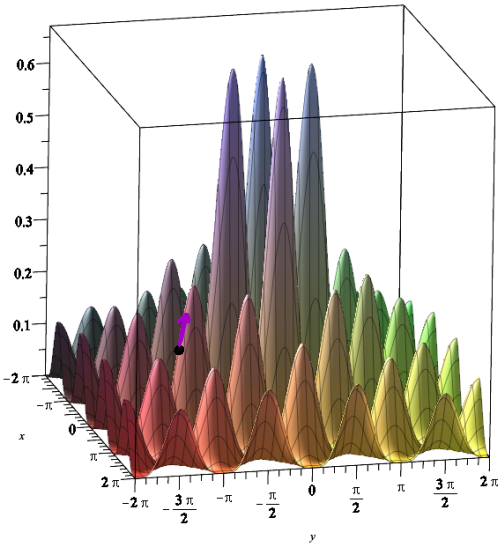
Find the max value of this function

on $x \in [-2\pi, 2\pi]$

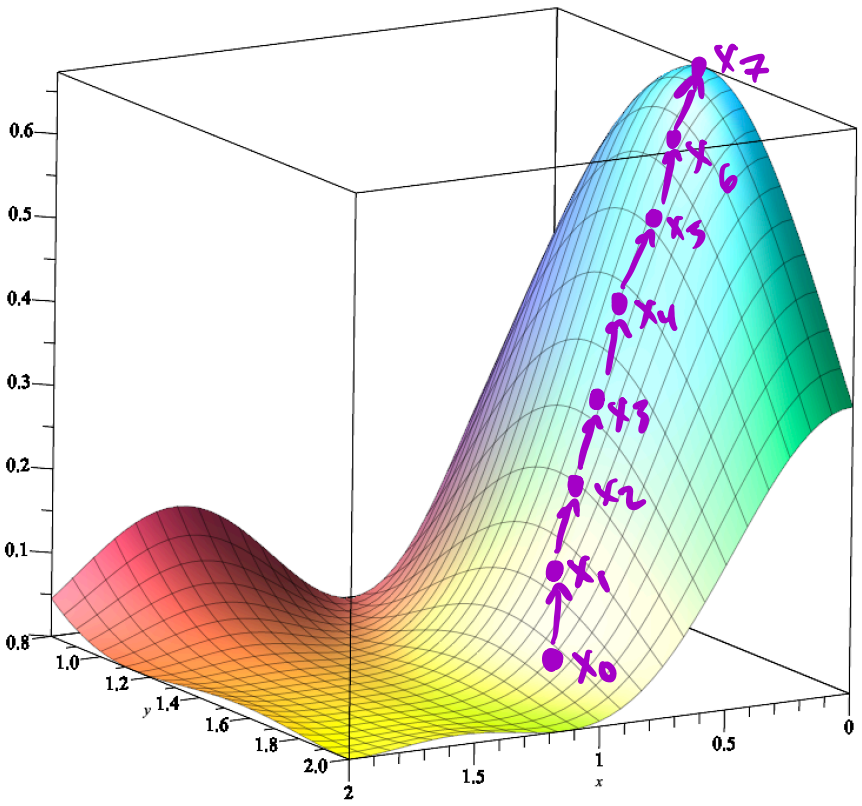
$$f(x,y) = \frac{\sin^2(x-y)\sin^2(x+y)}{\sqrt{x^2+y^2}} \qquad y \in [-2\pi, 2\pi]$$

Gradient Ascent (or Descent)

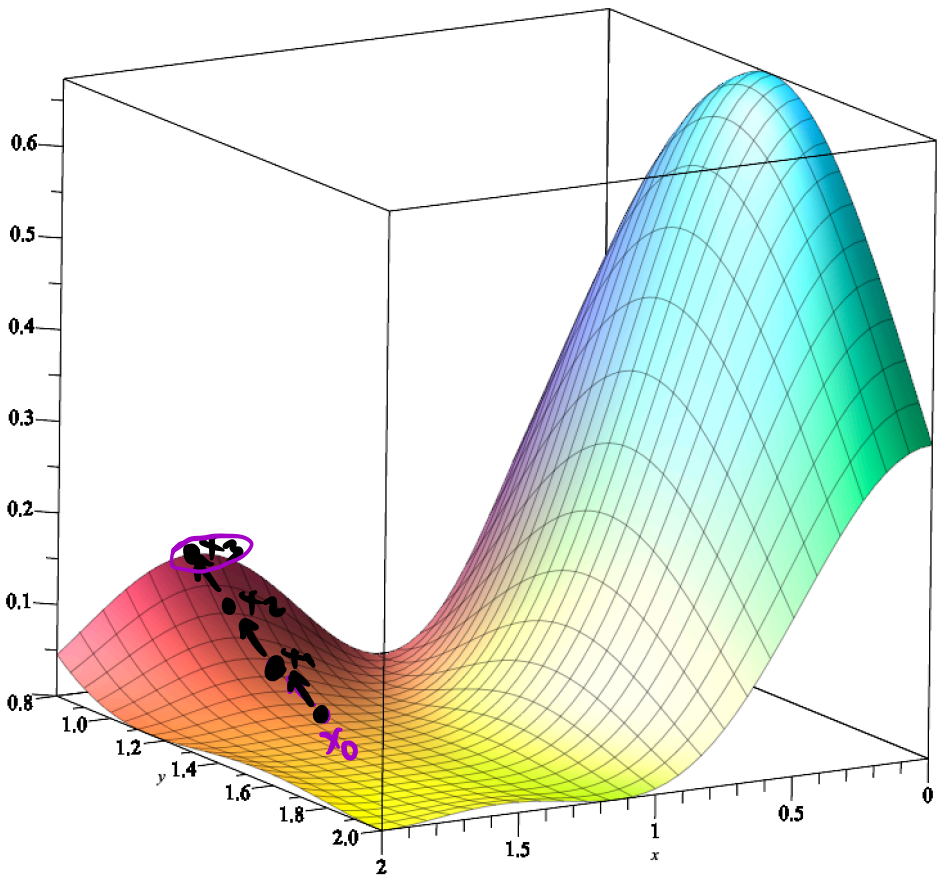
* Optimization method you learn in some math classes
* If you have a function $f(x,y)$ that is differentiable, you can compute the gradient at any point. The gradient is a vector that points you in the direction of steepest ascent.



Step 1) Start at any point
Step 2) Compute the gradient at that point
Step 3) Move a little bit in that direction
Step 4) Repeat steps 2 and 3 for a long time



We end up at the top of a hill.

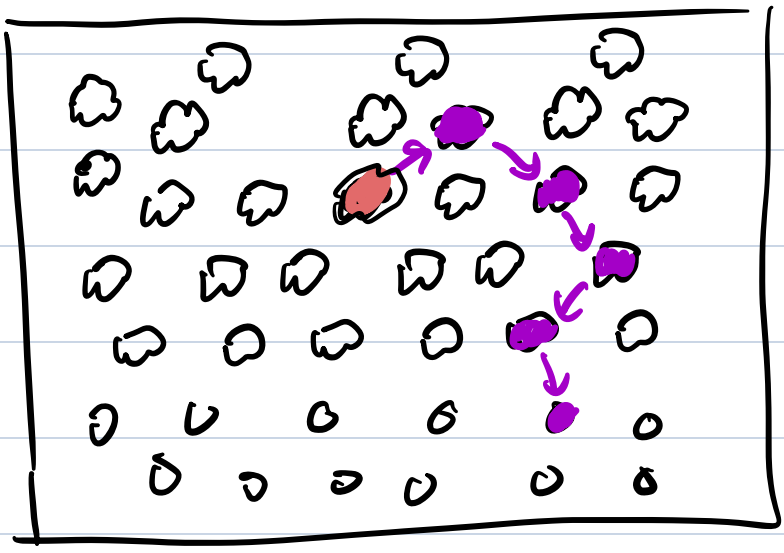


Gradient Ascent finds a local maximum (the top of some hill)

How can we simulate the idea of Gradient Ascent with a discrete search space?

[pretend you're in the mountains "standing at" one solution in the search space]

* look around you in a small radius at "nearby" solutions
* find the nearby solution with the best score
* go there and repeat



Ex: Traveling Salesman Problem
- search space: all tours on the input graph
- need a definition of "nearby" or "small radius"

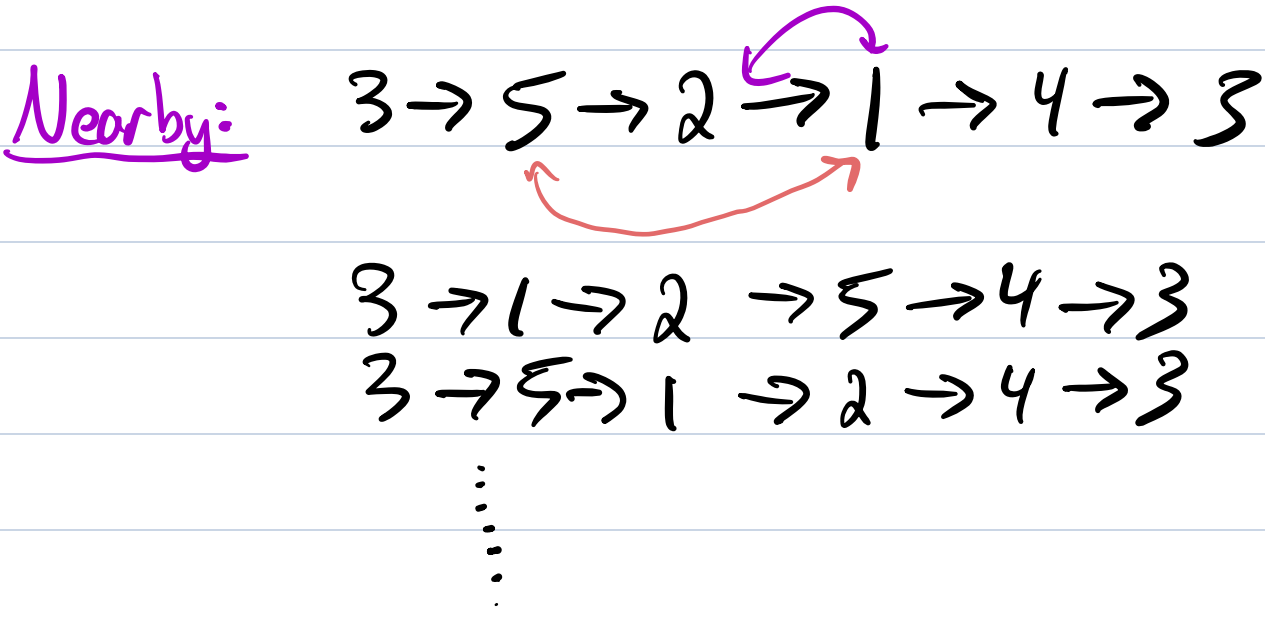
Ex: 5 cities, {1, 2, 3, 4, 5}
start at city 3

One element of the search space is:
$3 \to 5 \to 2 \to 1 \to 4 \to 3$

What other tours are "nearby"?
Up to you!
We're looking for tours that are very similar to this one.
"tweaks" = "small change"

One good example:
Nearby tours are the ones you get by swapping any two cities in the

starting tour (except the first/last one)



* start at a random tour
* calculate the score of all tours that can be formed by swapping two cities in the current tour
* move to the best one
* repeat

Same problem as gradient ascent: you probably get stuck in a local optimum.

Demos: TSP