Scientific Computing

March 26, 2025

Announcements

→ Homework 4 assigned, see D2L
Due Wednesday, April 2, 11:59pm

Today

→ Hill Climbing
→ Simulated Annealing

Office Hours:

Mon + Fri

9:30am - 10:30am

Cudahy 307

When $n=1$, this is just called "Hill Climbing"

MH #4: Hill Climbing

```
x = random element of S
while True:
    s = tweak(x)
    if score(s) > score(x):
        x = s
```

Repeat:

- Take a small step.
- If better, stay.
- If worse, go back.
- Repeat

HC = Hill Climbing

SA = Steepest Ascent

Demos: 09 - TSP HC Swap 2 50

10 - TSP HC Swap 2 300

11 - TSP HC RB 50

12 - TSP HC RB 300

RB = reverse a whole block of cities

Swap 2 = swap just 2 cities

50 cities		300 cities	
SA Swap 2	9.878	SA Swap 2	32.828
SA RB	6.487	SA RB	14.362
HC Swap 2	8.428	HC Swap 2	29.439
HC RB	6.412	HC RB	14.252

Regular HC beat Steepest Ascent

RB very much beat Swap 2

None of these four ever allow a worse score. You must always move uphill.

We'll talk a lot about exploration (diversification) vs. exploitation (intensification).

Looking in areas of the search space you haven't seen before

Searching the area you're already in for better and better solutions

Maximally exploitative: hill climbing
Maximally explorative: random search

We want things move in the middle, which means we have to allow sometimes going downhill!

MHs = "how to go downhill smartly"

Two ways we'll discuss for now:

(1) Random Restarts — "Hill Climbing with random restarts"

* Any H-C, or future MH, but after a while, stop and restart.

Example: MH #5 Hill Climbing with Random Restarts

```
best = random element of S
while True:
    x = random element of S
    for some amount of time:
        s = tweak(x)
        if score(s) > score(x):
            x = s
    if score(x) > score(best):
        best = x
```

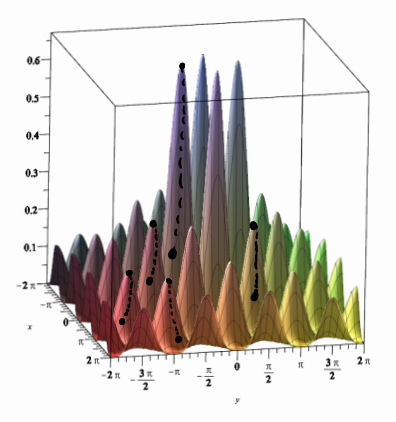
hill climbing

What does "some amount of time" mean?
Up to you. Possibilities: fixed length of time, or fixed # of iterations, or until no more improvement in some time, etc.

Demo: 13 - Contour HC w/ RR

Maple graphs

Demos 14, 15



Tweaking for continuous problems

Later: whole lecture on different ways to do this

Now: simple version

I'll state in terms of 2D $(x,y) \rightarrow z$

To tweak a current solution of (x,y)

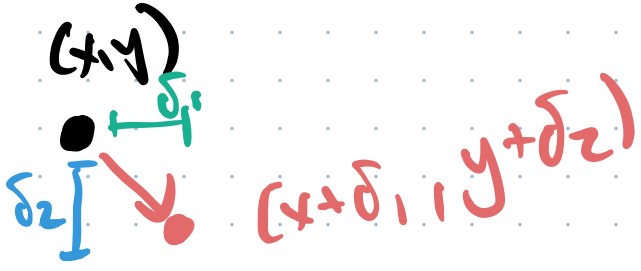
Pick δ_1 and δ_2 to be small random #s (pos. or neg.)

$$\text{tweak}((x,y)) = (x+\delta_1, y+\delta_2)$$

Ex:

δ_1 is random uniform from $[-0.01, +0.01]$

δ_2 is random uniform from $[-0.01, +0.01]$



At the start you might not know what the ranges should be!