Scientific Computing

Fri, March 20

Announcements

* Homework 4 due Friday, March 27
* Monday, April 6: No lecture, work from home day

Office Hours:

Mon, 9:30-10:30

Fri, 2:00-3:00

Cudahy 307

MH #1: Random Search

```
best = random element of S
while True:                    (quit whenever you want)
    x = random element of S
    if score(x) > score(best):
        best = x
```

Possible stopping conditions:

* best score does not improve for N iters
* preset number of iters
* you get impatient

Gradient Ascent inspires this next one.

MH #2: Steepest Ascent Hill-Climbing (Discrete only)

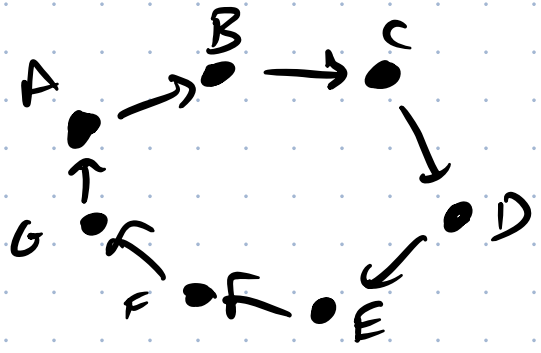
```
x = random element of S
while True:
    N = nbhd(x)
    s = element of N with the best score
    if score(s) > score(x):
        x = s
    else:
        quit
```

neighborhood (the set of things that are "close" to x)

best thing in neighborhood

Stopping conditions:
* run out of time
* no further improvement

How can we speed up scoring? Think about our tweak function. Suppose we have a tour:



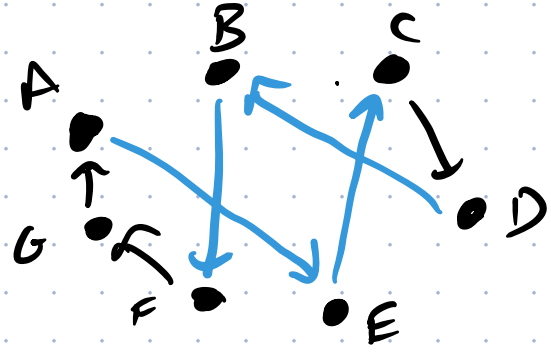
Let d = distance function.

$$\text{Score} = d(A,B) + d(B,C) + d(C,D) + d(D,E) + d(E,F) + d(F,G) + d(G,A)$$

Swap B and E:

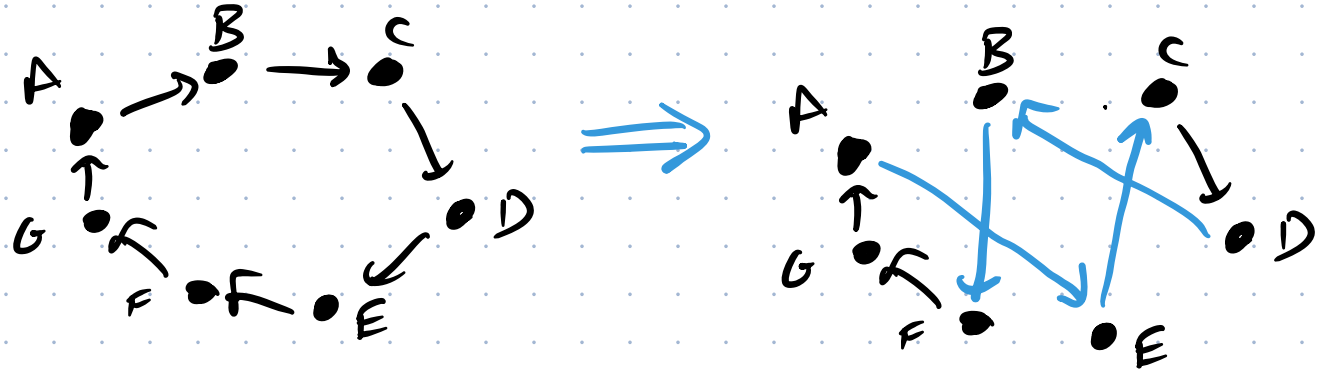
A → B → C → D → E → F → G → A

A → E → C → D → B → F → G → A



A → B → C → D → E → F → G → A

A → E → C → D → B → F → G → A



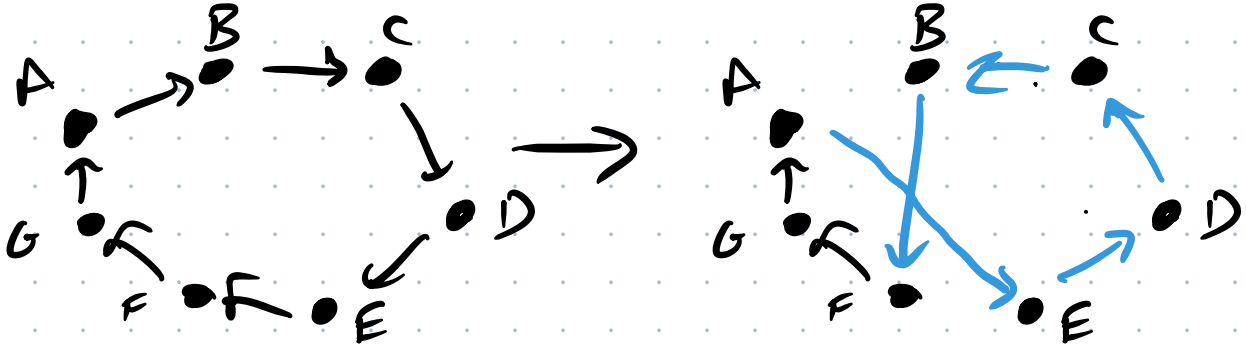
Four edges change.

$d(A,E)$ $d(E,C)$ $d(D,B)$

Score = ~~$d(A,B)$~~ + ~~$d(B,C)$~~ + $d(C,D)$ + ~~$d(D,E)$~~

+ ~~$d(E,F)$~~ + $d(F,G)$ + $d(G,A)$

$d(B,F)$



Assuming distance is symmetric yes.
This is picking two cities and reversing the whole block.

A → B → C → D → E → F → G → A

A → E → D → C → B → F → G → A

(faster scoring too!)

Demos: 06 - SA RB 50
07 - SA RB 300 08 / Fast Score

MH #3 n-Trial Steepest Ascent

↳ # of attempts per loop

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

nearby solution

Compute the score of n random neighbors and go to the best one

(if nothing beats x, it stays the same)

Later we will see good ways to tweak for continuous spaces.

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
```

Check 1 random neighbor (tweak)
if better, go there,
if not, stay and repeat

HC = Hill Climbing

SA = Steepest Ascent

RB = reverse a whole block of cities

Swap 2 = swap just 2 cities

Demos:

09 - TSP	HC	Swap 2	50
10 - TSP	HC	Swap 2	300
11 - TSP	HC	RB	50
12 - TSP	HC	RB	300

50 cities

SA	Swap 2	9.82
SA	RB	6.48
HC	Swap 2	8.42
HC	RB	6.45

300 cities

SA	Swap 2	32.82
SA	RB	14.36
HC	Swap 2	29.43
HC	RB	14.25

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

steepest ascent

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

* 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
```

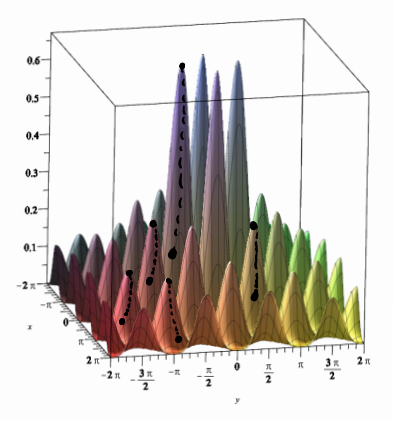
Hill-climbing once

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



(2) Probabilistic Hill-Climbing

Allow yourself to make a move that decreases your score, with some probability. More on this next time.

Topic 13 - Simulated Annealing

* Annealing Metal

[youtube video]