Scientific Computing

Wed, March 18

Announcements

* Homework 4 due Friday, March 27
* I'm still grading exams
* Monday, April 6: No lecture, work from home day

Office Hours:

Mon, 9:30-10:30
Fri, 2:00-3:00

Cudahy 307

Topic 12 - Hill Climbing

With Gradient Ascent as our inspiration, we want to think about ways to search for global optima in cases where our search space is (1) discrete
(2) cts, but can't compute a gradient

plural of "optimum"

maximum or minimum

Problem Setup:

* Search space S full of candidates

* Scoring function: score(x), $x \in S$
  (also called "fitness" w/ biological inspiration, or "quality")

how good or bad x is

* A way to generate either

- all the candidates near a candidate, the "neighborhood" nbhd(x)

probably doesn't make total sense in cts space

OR

- a random candidate near a candidate (sometimes called a "tweak") tweak(x).

"nearby" is up for you to define, and different definitions can totally change how a metaheuristic behaves.

Two running examples in this section.

(1) TSP:

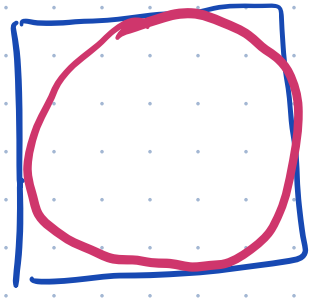
* discrete → finite search space
* score = cost of tour, want to minimize
* nbhd(x): suppose $x = c_1 \to c_2 \to \dots \to c_n \to c_1$
  Define the neighborhood to be all ways of picking two cities and swapping them (excluding c_1). $O(n^2)$
  size = $\binom{n-1}{2} = \frac{1}{2}(n-1)(n-2)$ (big!)

How large is the neighborhood?

* tweak(x): a random thing in the nbhd of x

(2) optimizing a continuous function in two variables $f(x,y)$.

* continuous
* score = value of the function
* nbhd(x) = all points within some fixed distance δ of x
* tweak(x) = a random point in nbhd(x)



δ small

infinite

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

This is not a good metaheuristic in most cases! It doesn't use any old information to guide future choices.

[2 Demos]
- 01: TSP random
- 02: Contour 1- random

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

best thing in neighborhood

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

What does this do? Climbs right up the hill you start on.

Pros

* Finds a local optimum

Cons

* Unlikely to find global optimum except in very nice spaces
* very slow, especially if nbhds are big, like TSP.

why?

Only really doing two things:

(1) generating the neighborhood

(2) scoring each element of it

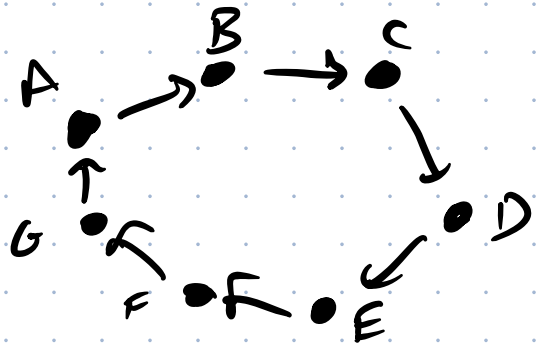
Scoring a tour with 300 cities is not horrible — 300 distance calculations (two subtractions, two squarings, one addition, one square root)

But bad when you do it $\binom{299}{2} = 44{,}551$ times.

Often, you don't have to restore a solution from scratch because it only changes a little bit. More on this in a bit.

Demos: 03 - TSP St. Asc. 50
04 - TSP St. Asc. 300 slow!

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



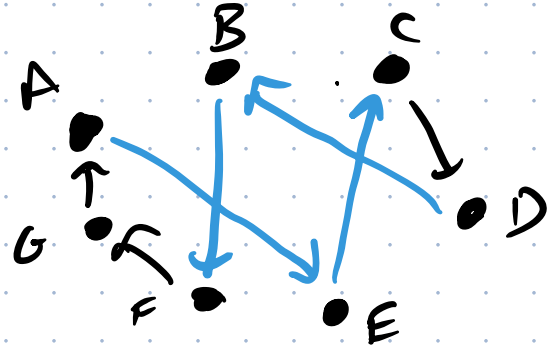
Let d = distance function.

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

Swap B and E:

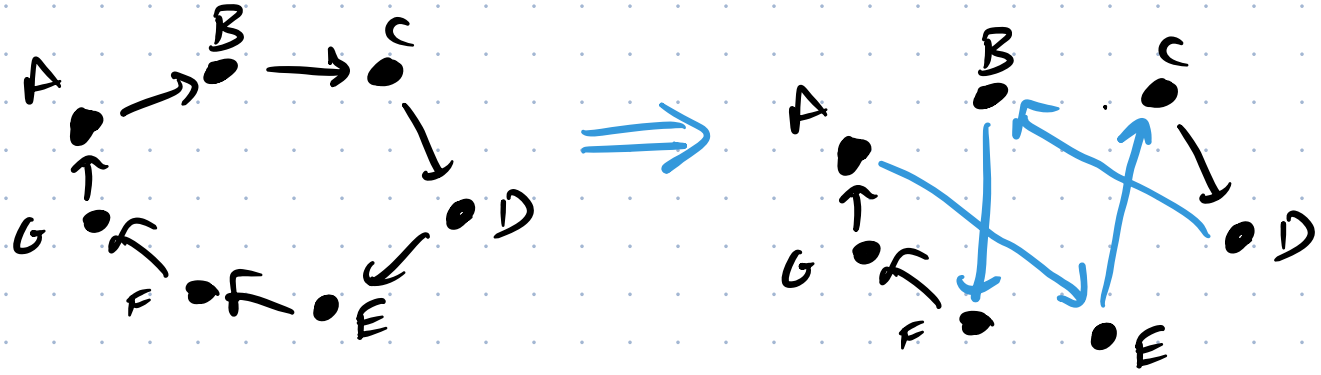
A → B → C → D → E → F → G → A

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

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

$d(B,F)$

If you have 300 cities, still only 4 edges change:

new score = old score - 4 edges + 4 edges.

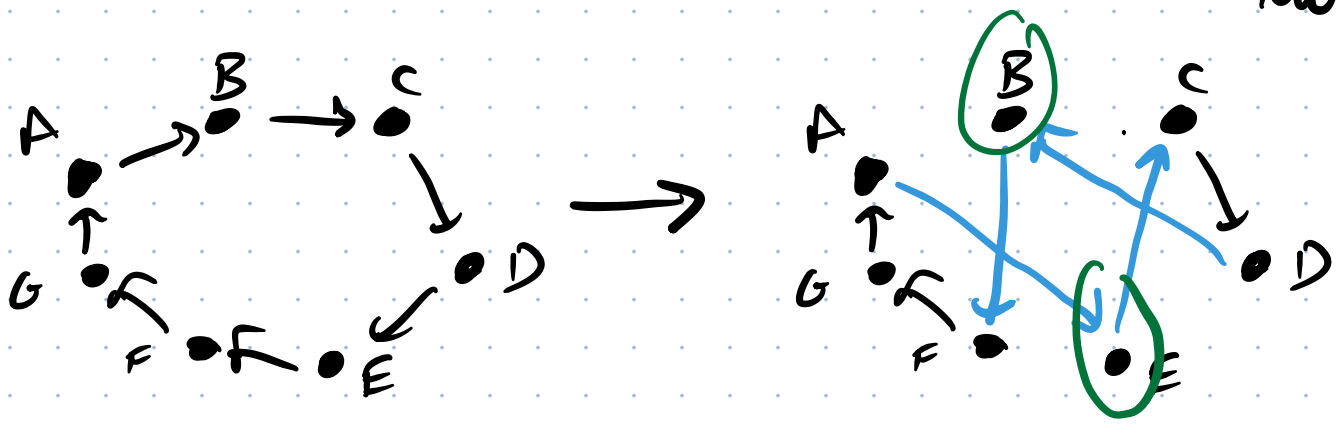
8 distances instead of 300 = $\frac{300}{8} = 37.5\times$ faster!

Demo 05

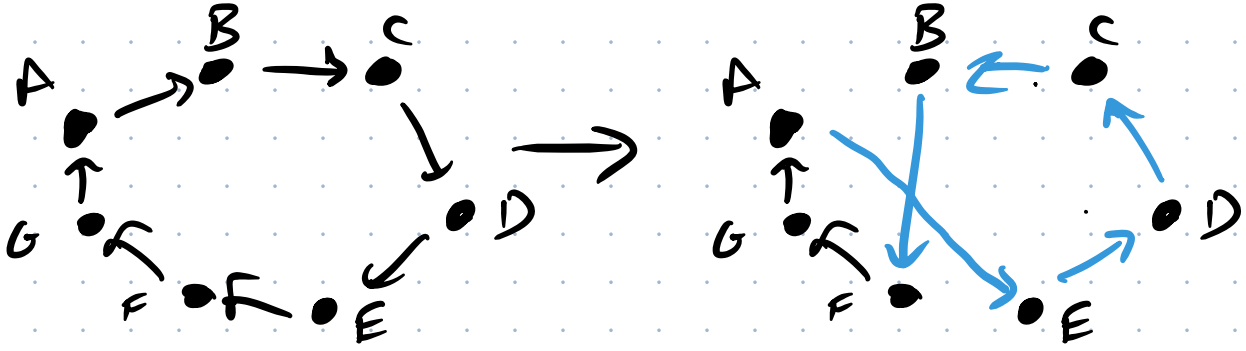
Second question: Is this a good tweak?

Our demos suggest maybe not.

Why not? Small tweaks are better (at least for now).



4 edges changed... can we change just 1?
No. How about 2?



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

The big theme of MHs is that they are super flexible. You can and should tweak them in all kinds of ways for particular problems. Always experiment.

How can we adapt this for continuous spaces?

How can we adapt this for continuous spaces?

Check n things in the neighborhood and take the best.

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

(if nothing beats x, it stays the same)

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