Wed, Apr. 3, 2024

Scientific Computing

Announcements

* Homework 4 due Fri (or Mon if you have asked for an extension)
* No in-person lecture on Mon, I will post a video

No O.H. on Monday.

Topic 11 - Hill-Climbing

RECAP

Problem Setup:

* Search space S of candidates
* Scoring function: score(x) for $x \in S$ (also called fitness or quality)
* A way to generate either:
  - all the candidates "near" a particular candidate

    the set of all candidates "near x" is called the "neighborhood of x"

    Notation: nbhd(x)

  OR

  - a random candidate "near" another one, a "tweak"

    tweak(x) = an element of the search space that is "near" x.

→ "nearby" is up for you to decide

→ different definitions can give better or worse solutions

Two running examples in this lecture:

(1) TSP:

* discrete
* search space = all tours of the cities
* score = sum of the distances traveled, we are minimizing
* Let $x = C_1 \to C_2 \to \dots \to C_n \to C_1$

Define nbhd(x) to be all tours you can get by swapping two cities.

How big is nbhd(x)?

A→B→C→D→E→A

A→D→C→B→E→A

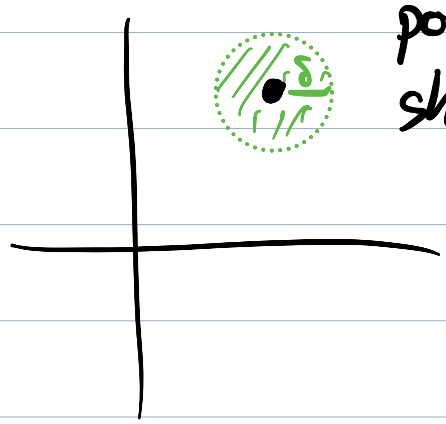
$$\binom{n-1}{2} = \frac{(n-1)\cdot(n-2)}{2} \approx n^2/2$$

* tweak(x): randomly pick two cities and swap the

↓

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

* continuous search space containing all (x,y) points, maybe in some bounds
* score = the value of the function at that point
* nbhd(p) = all points within some fixed Euclidean distance δ of p.



point is p
shaded is nbhd(p)

* tweak(p) = one random point in nbhd(p)

Metaheuristic #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
```

Stopping Conditions (many possible):

* best score hasn't improved for N iterations
* preset # of iterations to do
* you get bored
* if you have some sense ahead of time of what the optimal score should be, stop when you get within ε of that score

This not a good MH in most cases because it does not retain information to guide future choices.

[2 demos] { 01: TSP random
02: Contour-1 random

MH #2 - Steepest Ascent Hill-Climbing

inspired by gradient ascent (discrete only, otherwise $|N| = \infty$)

```
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
```

(circled ">" → $\geq$) what if a tie? up to you

May need stopping conditions like before

This mimics gradient ascent but for discrete spaces. It climbs right up to the top of a hill, then stops.

Pros	Cons
* Find a local optimum	* Unlikely to

* Fast-ish

find a <u>global</u> optimum except in very nice spaces
* Slow when nbhds are big.

→ Generating and scoring big neighborhoods

TSP with 300 cities

Scoring one element of the search space is not that bad

300 distance calculations

each is two subtractions
two squarings
one addition
one $\sqrt{\ }$

Size of nbhd: $\binom{299}{2} = 44{,}551$

Scoring 44,551 of them is slow.

Demos 3, 4