March 24 - Lecture #25

# Topic 12 - Hill Climbing

(1) TSP
(2) optimizing a continuous function

→ nbhd / tweak

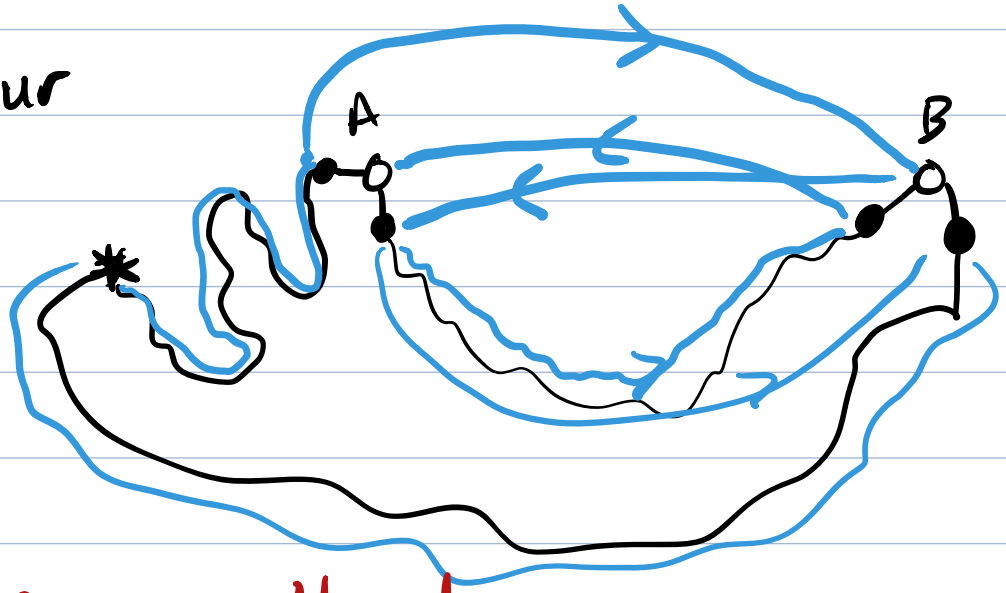
↳ pick any two cities (except the first) swap them

3 → 1 → 2 → 5 → 4 → 3

⇓

3 → 5 → 2 → 1 → 4 → 3

Tour




Swap A and B

Changing 4 edges

# MH #1: Random Search

# MH #2: Steepest Ascent Hill-Climbing inspired by Grad. Ascent - 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
```

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

Pros

* Find a local opt.

Cons

* Unlikely to find a global opt. unless you get lucky
* only discrete

* can be very slow

How can we adapt this to continuous space?

* discretize the space
* Check N things in the nbhd, and take the best.

MH #3: n-Trial Steepest Ascent

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

When n=1: called "Hill-Climbing"
try a random neighbor, if it's better, move there

MH #4 - Hill Climbing

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