Scientific Computing

March 31, 2025

Announcements

→ Homework 4 due Wednesday, 11:59pm

Today

→ Simulated Annealing

Office Hours:

Mon + Fri

9:30am – 10:30am

Cudahy 307

At the start, the system has a high temperature and the prob. of accepting a worse move is high.

Over time, the system cools down (lower temp.) and the probability of accepting a worse move decreases.

Very high temp: basically a random walk, accepting all tweaks

Very low temp: basically hill climbing

Technical Details:

* Assuming maximizing!

* **Acceptance Condition:**

Suppose the current temperature is T.

Let $s = \text{tweak}(x)$ and $\Delta = \text{score}(s) - \text{score}(x)$

If $\Delta > 0$, then s is an improvement over x, always accept

$\cancel{e^{1/T}}$

If $\Delta \leq 0$, then s is worse than x.

Accept with probability $p = e^{\Delta / T}$.

[$T > 0$, so $\Delta / T \leq 0$, so $0 < e^{\Delta / T} \leq 1$]

High $T \Rightarrow \Delta / T$ close to $0 \Rightarrow e^{\Delta / T}$ close to 1

Why this formula $p = e^{\Delta/T}$?

Comes from the Boltzmann distribution in physics.

(probability a system is in a certain state given its energy)

* If minimizing, add a negative sign $p = e^{-\Delta/T}$

$\boxed{\text{If minimizing, } p = e^{-\Delta/T}}$

* Cooling Schedule:

How the temperature changes over time.

- Geometric (most common)

  parameter α (decay rate)

  each time the temperature changes, we set

  $$T = T \cdot \alpha$$

Goes down 10% every time it changes

<u>Ex:</u> Initial Temp $T_0 = 10$, $\alpha = 0.9$

T goes $10 \to 9 \to 8.1 \to 7.29 \to \ldots$

(approaches 0)

The n^{th} temperature is $T_n = T_0 \cdot \alpha^n$

- Linear (less common)

  parameter β

  each time the temperature changes,

  we set $T = T - \beta$.

  The n^{th} temperature is $T_n = T_0 - \beta \cdot n$

- Many other possibilities, even non-monotone ones!

Process:

Pick an initial temperature T (How?)

x = random solution

best = x

Repeat:

For a while: (How long?) (How long?)

- $s = \text{tweak}(x)$
- $\Delta = \text{score}(s) - \text{score}(x)$
- if $\Delta > 0$: # improvement
  - $x = s$
  - if score(x) > score(best):
    - best = x
- else: # worse sol or =
  - r = random # in $[0,1]$
  - if $r < e^{\Delta/T}$:
    - $x = s$

— try one tweak

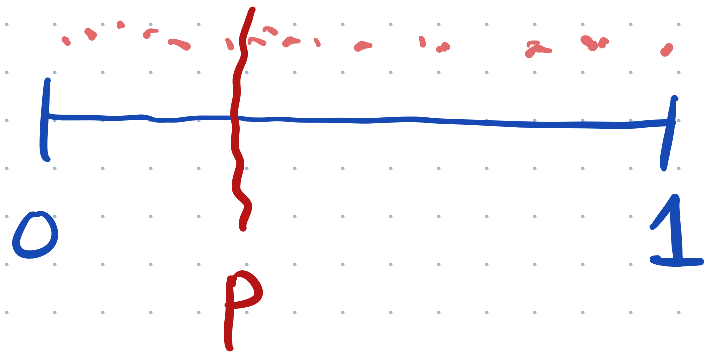
keep track of best sol ever seen

see next slide

adjust the temperature according to the (How?) cooling schedule

How do we go into an if statement with probability p?

Pick a uniform random # in the interval [0, 1]



20% of random #s in [0,1] will be less than 0.2

Example, $p = 0.2$ for 20%

if [random #] < p:
    do the thing

Questions to answer:

* How to pick initial temp
* How long to loop for each temp
* When to stop
* How to cool

art, not science

Picking the Initial Temp

First pick p_0: the initial probability with which you want worsening moves to be accepted.

What should it be? Opinions differ. Depends on the problem/landscape/etc. (art!)

↳ lots of little hills?
fewer huge hills?

$p_0 = 1.0$

$p_0 = 0.9$: safer, if you have plenty of time, very random at the start

$p_0 = 0.5$: good in many cases

$p_0 = 0.2$: sometimes good, especially in slow cases or landscapes with many short hills (it's easy to get down off short hills to get to new ones)

Suppose we want $p_0 = 50\% = 0.5$

Okay, how can we find the temp T that leads to p_0? Recall $\boxed{p = e^{\Delta/T}}$ But we don't know Δ.

know (p, e) — don't know (Δ) — want to solve (T)

We will approximate the average value of Δ, then use that: "On average, when a tweak is worse, how much worse is it?"

```
trials = []
while len(trials) < 1000:      (or whatever)
    x = random solution
    s = tweak(x)
    if score(s) < score(x):    (worsening)
        trials.append(score(s) - score(x))
avg = sum(trials)/1000
```

average amount of "worse" a worse solution is

Now we have an average Δ for a worse solution

$$p = e^{\Delta / T} \Rightarrow \ln(p) = \Delta / T \Rightarrow T = \Delta / \ln(p)$$

So, a good starting temperature T_0 to force a particular p_0 is:

$$T_0 = \frac{\text{avg}\,\Delta}{\ln(p_0)}$$

This far from exact:

(1) We just took an average

(2) The avg. worsening in the context of running SA (going up and down) might be very different than the avg. worsening from a random solution.

While you're running Simulated Annealing, print the % of worsening solutions accepted at each temperature, and then you can adjust T_0 as needed.

How long do we run at a fixed temp before moving to the next temp?

$N = 1000$ if not too slow

Some possibilities:

* N tweak attempts in total
* K worsenings rejected or L worsenings accepted, whatever comes first

(ex: 1000 accepted or 75000 rejected)

How long do we cool the system before stopping

* Run out of time
* No worsening moves accepted in a while (basically hill-climbing)
* Pre-set end temperature $T_f = 0.001 \cdot T_0$.

Code Review + Demos

* #13, #12, no GUI
* #5-8, TSP

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

HC = Hill Climbing
SA = Steepest Ascent

RB = reverse a whole block of cities
Swap 2 = swap just 2 cities