Note Title

Simulated Annealing

Motivated by "annealing" process

liquid cooling, plate @ high temperature → form crystals

- if liquid cools too quickly –
  too slowly – } improper formation of crystals

Key: lower temperature at a rate that encourages proper formation

---

Applied to Hill-Climbing

<u>Problem</u>: Hill-Climbing can get stuck @ local optima
yet is computationally advantageous otherwise

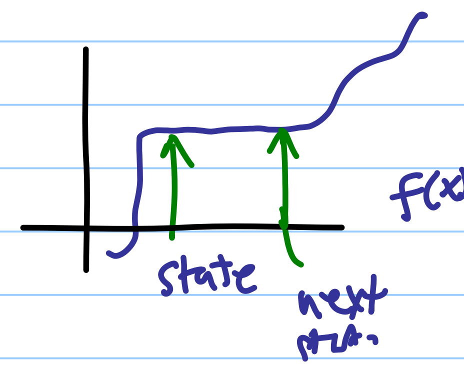
Question: How to "relax" hill-climbing enough to allow it to "skip away" from local optima?

Basic Hill-Climbing

do this MAX times, w/random choices of Initial state

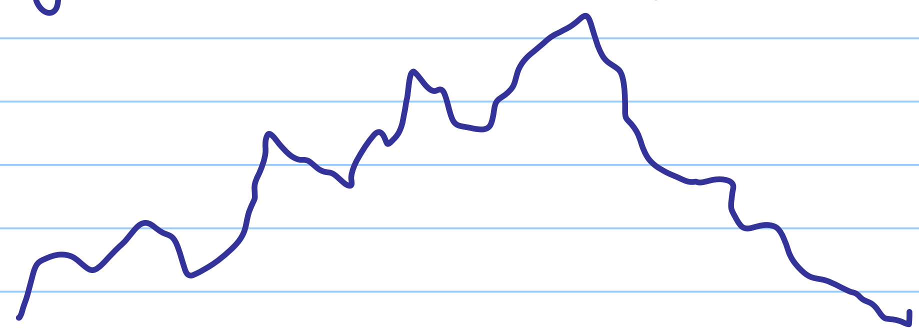
```
state ← Initial state
value ← f(state)

while not goal(state)
    M ← ApplicableRules(state)

    r ← arg max   { f(ApplyRule(m, state)) }
          m∈M

    nextstate ← ApplyRule(r, state)
    if( f(nextstate) < f(state)) STOP // local optimum
    else
        state ← nextState
```

Q: suppose f(nextstate) == f(state) ? Continue ?



IDEA: suppose $f(\text{next}) > f(\text{state}) - \epsilon$ { i.e, almost as good as f(state)

maybe we should keep going —



Track behavior over a longer trend —

Devise a hill-climbing function RELATED TO objective function

And be willing to accept temporary setbacks —

This function should (a) bias towards improvements in f()

(b) allow occasional setbacks

(c) use probability

Select between candidates for next state using prob' method

IDEA: like "proportional fitness" in selecting breeding pairs in genetic alg's —

f_1 f_2 .... f_n for elements of M = neighbors of state

Select randomly —

Note: Scaling may be helpful here

IDEA 2: (Simulated Annealing)

Define a Temperature value $\underline{\underline{T}}$ that guides the amount of tolerance we have for fluctuations between good + bad sol'ns

lower T gradually —

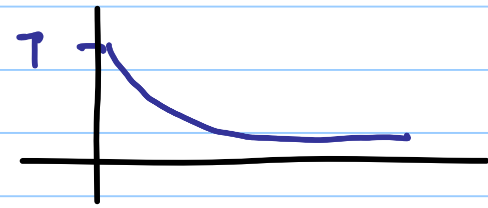
i.e., keep dividing by 2: $T, T/2, T/4, \ldots$

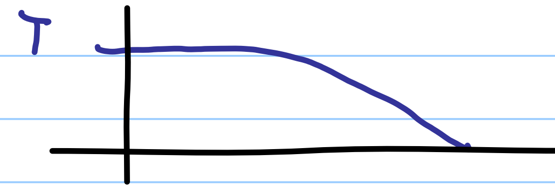
Keep dividing by $r > 1$:

$T, T/r, T/r^2, \ldots$ i.e., $r = 3/2$

$T, 2T/3, 4T/9, \ldots$

or logarithmic/exponential





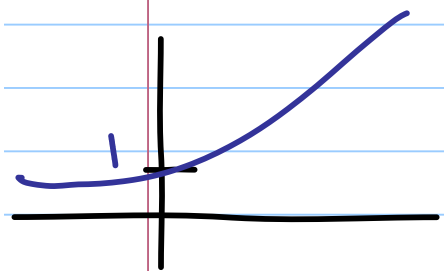
Given state V_c (current state) $f(V_c)$ current obj fn value

neighbor V_n (neighboring state) $f(V_n)$

maximize $f()$ (in general, want $f(V_n) > f(V_c)$)

Compute $p(V_n) = \dfrac{1}{1 + e^{\frac{[f(V_c) - f(V_n)]}{T}}}$

$\boxed{T = \text{Temperature}}$



Suppose

$$\begin{cases} f(v_c) = f(v_n)\,? & \text{Then } P(v_n) = \dfrac{1}{1+e^0} = \dfrac{1}{2} \\[2ex] f(v_c) < f(v_n)\,? & \text{Then } P(v_n) = \dfrac{1}{1+e^{-\alpha}} \text{ for some } \alpha > 0 \\ & e^{-\alpha} < 1\ ? \quad P(v_n) > \dfrac{1}{2} \\[2ex] f(v_c) > f(v_n) & P(v_n) = \dfrac{1}{1+e^{\alpha}} \quad e^{\alpha} > 1 \quad p(v_n) < \dfrac{1}{2} \end{cases}$$

$P(v_n)$ is the probability of selecting v_n as next solution —

compute $r = \text{rand}(0,1)$

if $(r < P(v_n))$

choose v_n — otherwise, continue

Simulated Annealing:

run this MAX times

$T = T_{max}$

V_c = random starting point

M = successors(V_c)

pick $v_n \in \{M\}$

if($f(v_n) > f(v_c)$)

  $v_c \leftarrow v_n$

else if ($f(v_n) \leq f(v_c)$)

  $r \leftarrow$ Random$(0,1)$

  if $\left(r < \dfrac{1}{1 + \exp((f(v_c) - f(v_n))/T)} \right)$

    $v_c \leftarrow v_n$

decrease T

loop until choosing new v_c or N tries