CS 540 Introduction to Artificial Intelligence

Search III: Optimization

University of Wisconsin–Madison

December 1, 2025

Fall 2025

Announcements

- **Homework**:
  - HW9 due on **Tuesday Dec 2 at 11:59 PM**
  - HW10 released Dec 2, due Dec 9 at 11:59 PM

- Final Exam
  - Saturday, Dec 13, 12:25-2:25 PM
  - CHEM S249

Advanced Search
Ethics and Trust in AI

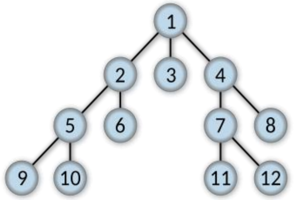
Outline

- Advanced Search & Hill-climbing
  - More difficult problems, basics, local optima, variations
- Simulated Annealing
  - Basic algorithm, temperature, tradeoffs

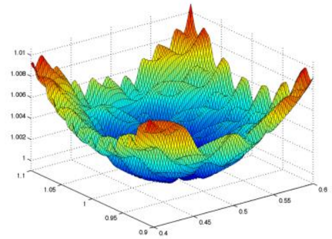
Search vs. Optimization

Before: wanted a **path** from start state to goal state

- Uninformed search, informed search



Wiki



TuringFin

New setting: optimization

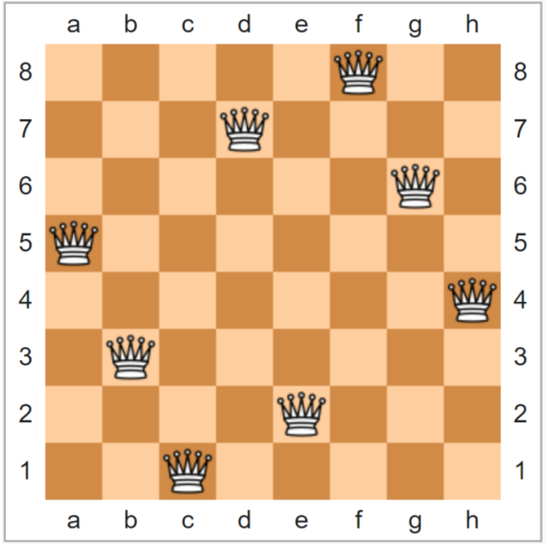
- States *s* have values ***f***(*s*)
- Want: Find *s* with optimal value ***f***(*s*) (i.e, **optimize** over states)
- Challenging settings: **too many states** for previous search approaches, but maybe not a differentiable function for gradient descent.

Examples: n Queens

A classic puzzle:

- Place 8 queens on 8 x 8 chessboard so that no two have same row, column, or diagonal.
- Can generalize to n x n chessboard.

- What are states *s*? Values ***f***(*s*)?
  - State: configuration of the board
  - ***f***(*s*): # of non-conflicting queens



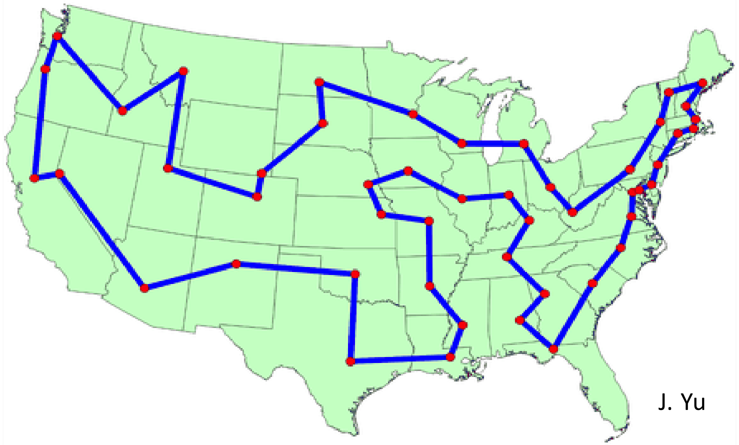
Wiki

Examples: TSP

Famous graph theory problem.

- Get a graph G = (V,E). **Goal**: a path that visits each node exactly once and returns to the initial node (a **tour**).
  - State: a particular tour (i.e., ordered list of nodes)
  - ***f***(*s*): total weight of the tour

(e.g., total miles traveled)



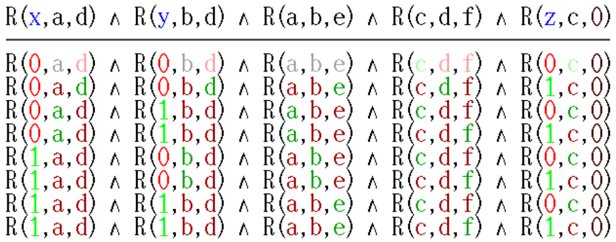
Examples: Satisfiability

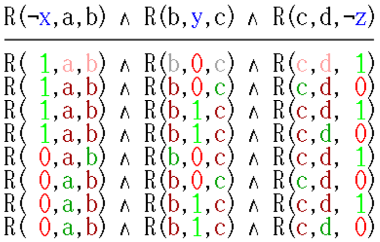
Boolean satisfiability (e.g., 3-SAT)

- Recall our logic lecture. Conjunctive normal form

$$(A \lor \neg B \lor C) \land (\neg A \lor C \lor D) \land (B \lor D \lor \neg E) \land (\neg C \lor \neg D \lor \neg E) \land (\neg A \lor \neg C \lor E)$$

  - Goal: find if satisfactory assignment exists.
  - State: assignment to variables
  - ***f***(*s*): # satisfied clauses





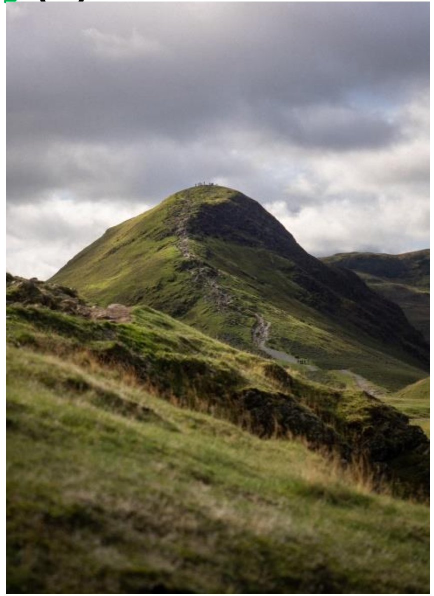
Wiki

Hill Climbing

One approach to such optimization problems

- Basic idea: start at one state, move to a neighbor with a better *f*(*s*) value, repeat until no neighbors have better *f*(*s*) value.

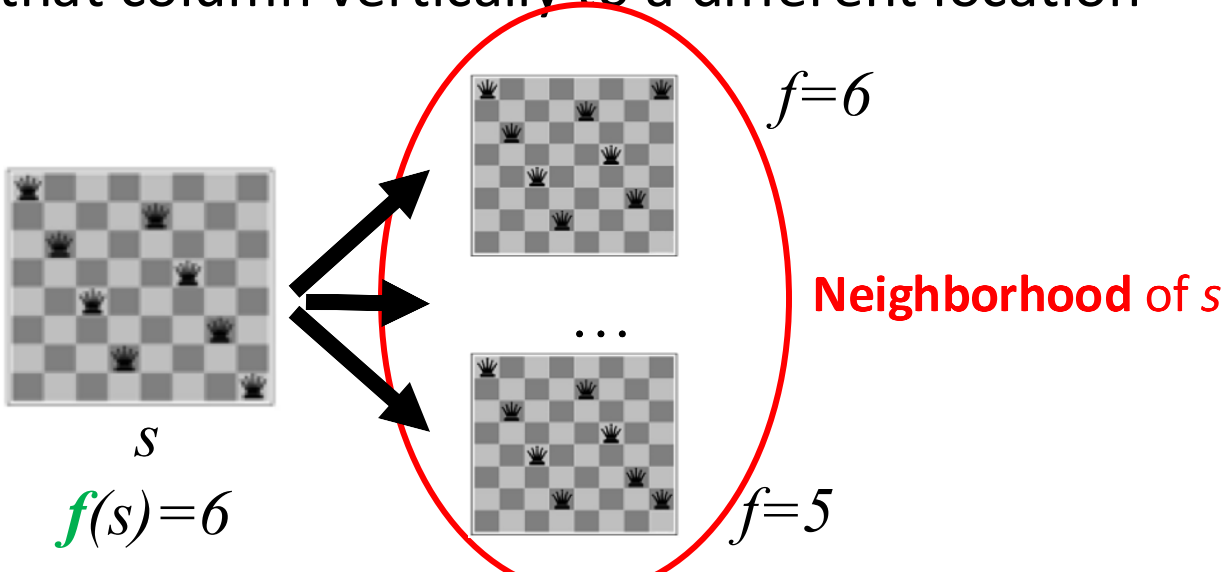
- **Q**: how do we define **neighbor**?
  - Not as obvious as our successors in search
  - Problem-specific
  - As we'll see, needs a careful choice



Defining Neighbors: n Queens

In n Queens, a simple possibility:

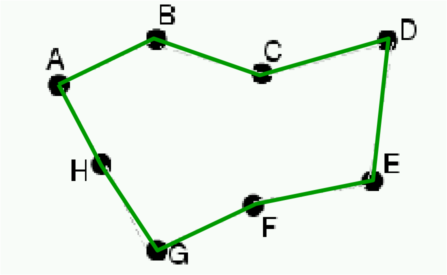
- Look at the **most-conflicting column** (ties? right-most one)
- Move queen in that column vertically to a different location



Defining Neighbors: TSP

For TSP, can do something similar:

- Define neighbors by small changes
- Example: 2-change: A-E and B-F



A-B-C-D-E-F-G-H-A

flip

A-E-D-C-B-F-G-H-A

Defining Neighbors: SAT

For Boolean satisfiability,

- Define neighbors by flipping one assignment of one variable

Starting state: (A=T, B=F, C=T, D=T, F=T)

(A=**F**, B=F, C=T, D=T, E=T)
(A=T, B=**T**, C=T, D=T, E=T)
(A=T, B=F, C=**F**, D=T, E=T)
(A=T, B=F, C=T, D=**F**, E=T)
(A=T, B=F, C=T, D=T, E=**F**)

$A \vee \neg B \vee C$
$\neg A \vee C \vee D$
$B \vee D \vee \neg E$
$\neg C \vee \neg D \vee \neg E$
$\neg A \vee \neg C \vee E$

Hill Climbing Neighbors

Q: What's a **neighbor?**

- **Vague definition:** for a given problem structure, neighbors are states that can be produced by a small change
- **Tradeoff**!
  - Neighborhood too small? Will get stuck.
  - Neighborhood too big? Not very efficient

- **Q**: how to pick a neighbor? Greedy
- **Q**: terminate? When no neighbor has better value

Hill Climbing Algorithm

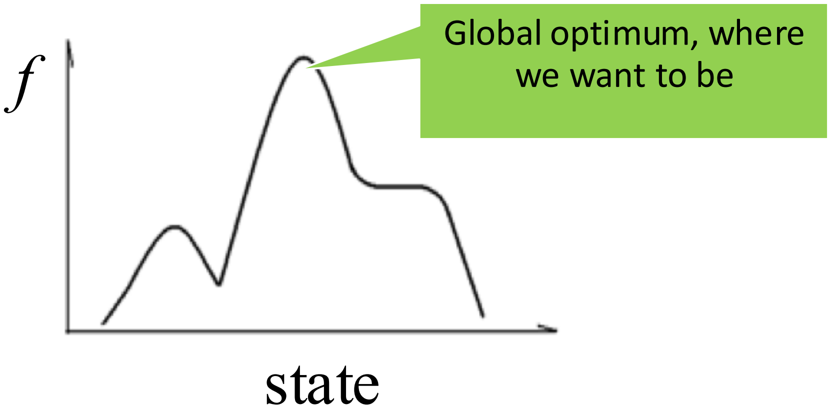
Pseudocode:

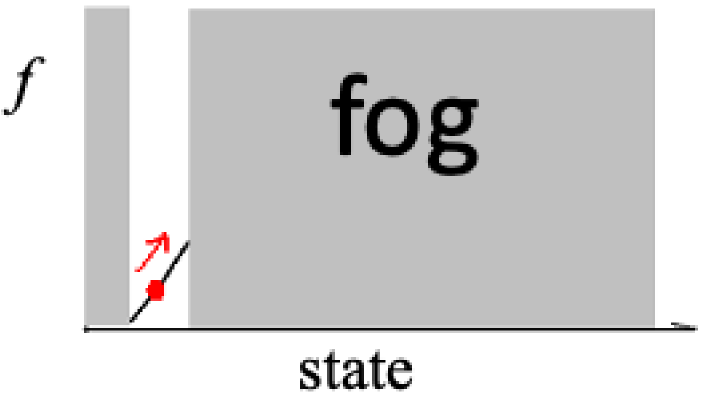
1. Pick initial state s
2. Pick t in **neighbors**(s) with the best $\boldsymbol{f}(t)$
3. if $\boldsymbol{f}(t)$ is not better than $\boldsymbol{f}(s)$ THEN stop, return s
4. $s \leftarrow t$. goto 2.

What could happen? **Local optima!**

Hill Climbing: Local Optima

Q: Why is it called hill climbing?



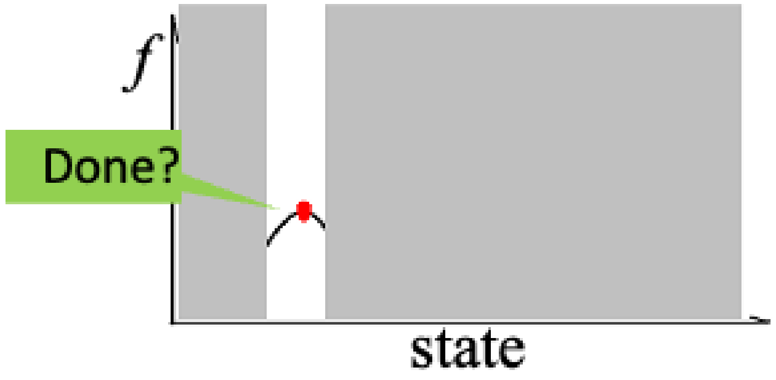


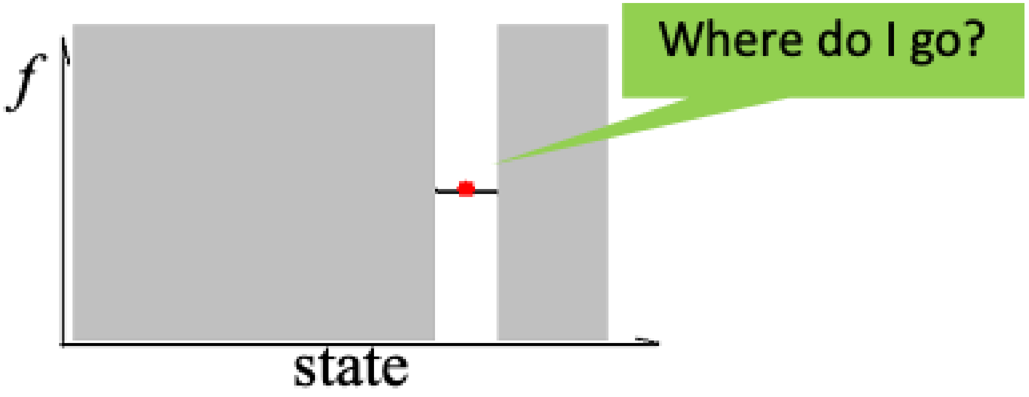
L: What's actually going on.

R: What we get to see.

Hill Climbing: Local Optima

Note the **local optima**. How do we handle them?





Escaping Local Optima

Simple idea 1: random restarts

- Stuck: pick a random new starting point, re-run.
- Do *k* times, return best of the *k* runs.

Simple idea 2: reduce greed

- “Stochastic” hill climbing: randomly select between neighbors.
- Probability of selecting a neighbor should be proportional to the value of that neighbor.

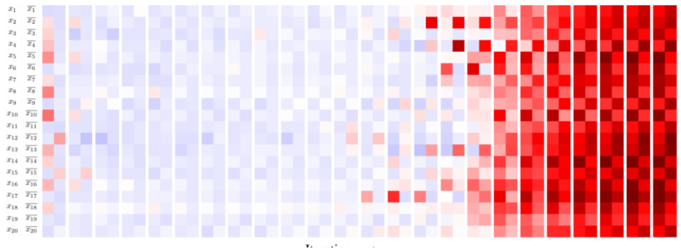
Hill Climbing: Variations

Q: neighborhood too large?

- Generate random neighbors, **one at a time**. Take the better one.

Q: relax requirement to always go up?

- Often useful for harder problems



D. Selsam

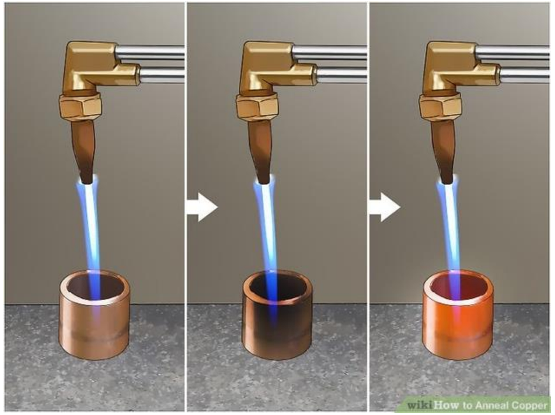
Simulated Annealing

A more sophisticated optimization approach.

- **Idea**: allow some downhill moves at first, then be pickier over time
- Pseudocode:

Pick initial state s; T=1
For k = 0 through K:
    $T \leftarrow T*0.99$ *(cool down)*
    Pick a random neighbour $t \leftarrow$ neighbor(s)
    If $f(t)$ better than $f(s)$, then $s \leftarrow t$
    Else with prob. $P(f(s), f(t), \boldsymbol{T})$ still do $s \leftarrow t$
Output: the best state ever seen

The interesting bit



wikihow.com

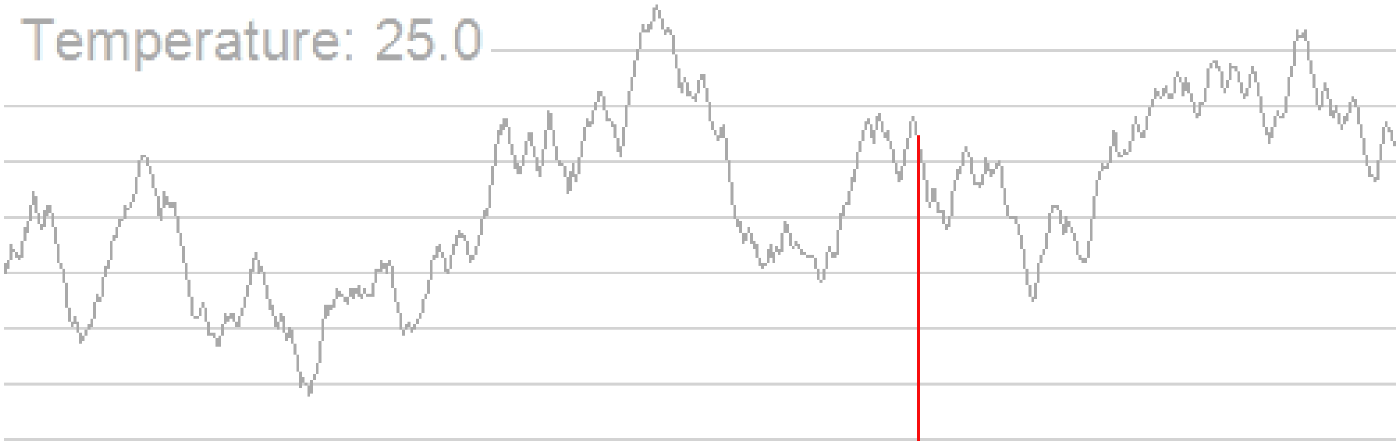
Simulated Annealing: Picking Probability

How do we pick probability P? Note 3 parameters.

- Decrease with time
- Decrease with gap $|f(s) - f(t)|$: $\exp\left(-\frac{|f(s)-f(t)|}{Temp}\right)$
- Temperature cools over time.
  - So: high temperature, accept any *t*
  - But, low temperature, behaves like hill-climbing
  - Still, $|f(s) - f(t)|$ plays a role: if big, replacement probability low.

Simulated Annealing: Visualization

What does it look like in practice?



Wiki

Simulated Annealing: Picking Parameters

- Have to balance the various parts., e.g., cooling schedule.
  - Too fast: becomes hill climbing, stuck in local optima
  - Too slow: takes too long.
- Combines with variations (e.g., with random restarts)
  - Probably should try hill-climbing first though.
- Inspired by cooling of metals
  - We'll see one more alg. inspired by nature