LECTURE 11

CS 4: INTRODUCTION TO ARTIFICIAL INTELLIGENCE

1

PREVIOUSLY . . .

- Previously, we cared about the paths to get from the initial state to a goal
- But what if we only care about getting to a goal?

2

LOCAL SEARCH ALGORITHMS

SEARCHING FROM A START STATE TO NEIGHBORING STATES

WITHOUT KEEPING TRACK OF THE PATHS

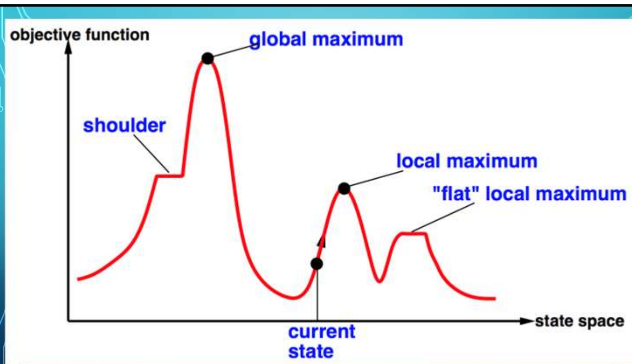
NOR THE SET OF STATES THAT HAVE BEEN REACHED

3

LOCAL SEARCH: PROS AND CONS

Pros	Cons
• Use very little memory • Can often find reasonable solutions in large or infinite state spaces • Can be used to solve optimization problems	• Not systematic

4



5

HILL-CLIMBING SEARCH

- The **hill-climbing search** keeps track of just the current state
  - Move to the best neighboring state
  - Stop when no neighbors are better
- The algorithm moves in the direction of **steepest ascent**
- Can use heuristics with this algorithm

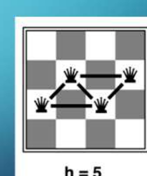
6

```
function HILL-CLIMBING(problem) returns a state that is a local maximum
    current ← problem.initial
    while true do
        neighbor ← a highest-valued successor state of current
        if VALUE(neighbor) ≤ VALUE(current) then
            return current
        current ← neighbor
```

7

n-QUEENS PROBLEM

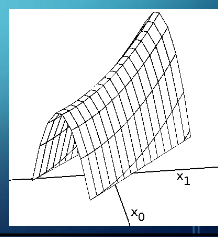
- Environment: a chess board of size $n \times n$ with n queens
  - $n = 1$ or $n \geq 4$
- State: n queens on the board, one per column
- Actions: move a queen in its column
- Goal: n queens on the board with no conflicts
- Heuristic: number of conflicts



8

HILL-CLIMBING SEARCH: PERFORMANCE

- Hill climbing is also known as **greedy local search**
- It is quick to make progress towards a solution
- However, it often gets stuck somewhere
  - Local maxima, ridges, plateaus



9

HILL-CLIMBING SEARCH: IMPROVEMENTS

- When stuck on a plateau, allow for a **sideways move** in hopes that the plateau is a shoulder
  - Cap the number of sideways move allowed
- Increases the success percentage, but also increases the number of steps to a solution

10

HILL-CLIMBING SEARCH: VARIATIONS

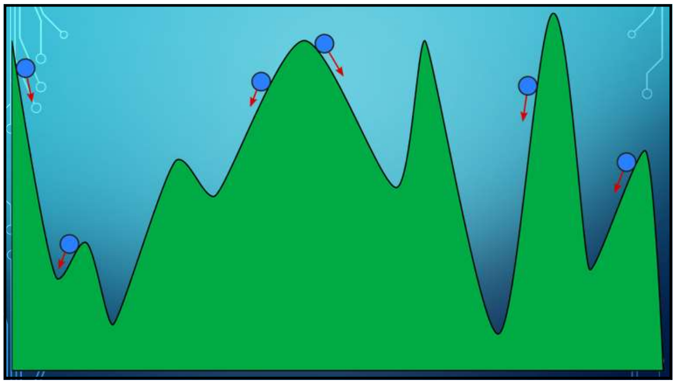
- **Stochastic hill climbing** chooses at random from among the uphill moves
- **First-choice hill climbing** generates successor states in random order, stopping when it finds one that's better than the current state
- **Random-restart hill climbing** conducts a series of hill-climbing searches from randomly generated initial states until a goal is found

11

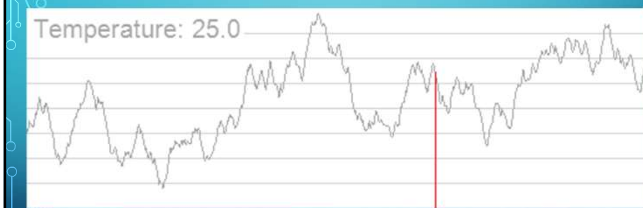
SIMULATED ANNEALING

- The **simulated annealing** algorithm is a version of stochastic hill climbing that allows for downhill moves
- Inspiration: the annealing process for metals and glass
  - Heat the material to a high temperature
  - Gradually cool it, allowing the material to reach a low-energy crystalline structure
- Perspectives: can be used for steepest ascent or for **gradient descent**

12



13



14

```
function SIMULATED-ANNEALING(problem, schedule) returns a solution state
    current ← problem.initial
    for t = 1 to ∞ do
        T ← schedule(t)
        if T = 0 then
            return current
        next ← a randomly selected successor of current
        ΔE ← VALUE(current) - VALUE(next)
        if ΔE > 0 then
            current ← next
        else
            current ← next only with probability e^(ΔE/T)
```

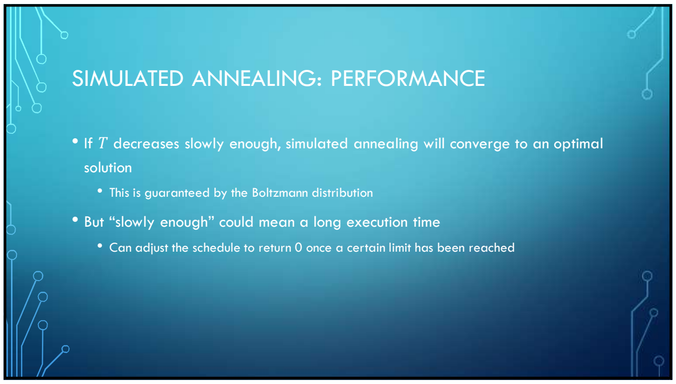
Boltzmann distribution

15

SIMULATED ANNEALING: TEMPERATURE SCHEDULE

- A common temperature schedule to use comes from Newton's law of cooling
  - "The rate of heat loss of a body is directly proportional to the difference in the temperatures between the body and its environment"
- $S(t) = ke^{-r}$
  - k is the initial temperature
  - r is the "coefficient of heat transfer" (smaller number means slower decay)

16


SIMULATED ANNEALING: PERFORMANCE
• If T decreases slowly enough, simulated annealing will converge to an optimal solution
• This is guaranteed by the Boltzmann distribution
• But "slowly enough" could mean a long execution time
• Can adjust the schedule to return 0 once a certain limit has been reached

17