Simulated annealing search

- Hill-Climbing that *never* makes “downhill” moves toward states with lower value (or higher cost) is guaranteed to be incomplete, because it can get stuck in local maximum.
- In contrast, purely random walk (moving to a successor chosen uniformly at random from set of successor) is complete but extremely inefficient
- How about combining the two? ➔ Simulated annealing, a version of stochastic hill climbing where *some downhill moves* are allowed: they are accepted readily early in annealing schedule and less often as time goes on.
-

Simulated annealing

- Origin; metallurgical annealing (high T to harden metals, then gradually cooling them)
- Switch point of view from hill climbing to **gradient descent** (i.e. minimize cost)
- **Idea:** escape local minima (or local maxima experienced with hill-climbing) by allowing some "bad" random moves
  - but gradually decrease their frequency
- Bouncing ball analogy:
  - Goal: get ball in deepest crevice of bumpy surface
  - If let ball roll, might get stuck in local minimum
  - If shake surface hard (high temperature), ball bounces out of LOCAL min, but if shake too hard, ball will be dislodged from GLOBAL min

  ➔ Best start to shake hard, then gradually reduce intensity (lower the temperature),
- Can prove: If T decreases slowly enough, best state is reached.
- Applied for VLSI layout in 1980s, airline scheduling, etc.

Simulated annealing

function SIMULATED-ANNEALING(*problem*, *schedule*) **returns** a solution state
  inputs: *problem*, a problem
    schedule, a mapping from time to "temperature"
  local variables: T, a "temperature" controlling the probability of downward steps

  current ← MAKE-NODE(*problem*.INITIAL-STATE)
  for $t = 1$ **to** ∞ **do**
    $T \leftarrow schedule(t)$
    if $T = 0$ **then return** *current*
    next ← a randomly selected successor of *current*
    $\Delta E \leftarrow next.\text{VALUE} - current.\text{VALUE}$
    if $\Delta E > 0$ **then** *current* ← *next*
    else *current* ← *next* only with probability $e^{\Delta E/T}$

Figure 4.5 The simulated annealing algorithm, a version of stochastic hill climbing where some downhill moves are allowed. Downhill moves are accepted readily early in the annealing schedule and then less often as time goes on. The *schedule* input determines the value of T as a function of time.

Properties of simulated annealing search

- One can prove: If *T* decreases slowly enough, then simulated annealing search will find a global optimum with probability approaching 1

- Widely used in VLSI layout, airline scheduling, etc.

Steepest Descent

1) S ← initial state
2) Repeat:
   a) $S' \leftarrow \arg\min_{S' \in \text{SUCCESSORS}(S)} \{h(S')\}$
   b) if GOAL?(S') return S'
   c) if h(S') < h(S) then S ← S' else return failure

Similar to:

- hill climbing with –h
- gradient descent over continuous space

Application: 8-Queen

Repeat n times:

1) Pick an initial state S at random with one queen in each column
2) Repeat k times:
   a) If GOAL?(S) then return S
   b) Pick an attacked queen Q at random
   c) Move Q in its column to minimize the number of attacking queens → new S [min-conflicts heuristic]
3) Return failure