Lecture 10: Local Search

CSCI 404 Artificial Intelligence, Spring 2018

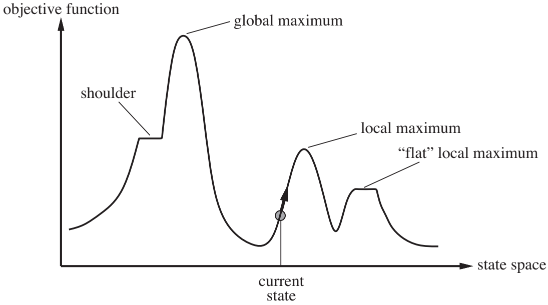
Hua Wang, Ph.D.

Department of Computer Science
Colorado School of Mines

February 13, 2018

Local search

- **Pros** : Hill climbing search (as well as local search) operates only one node of the search tree — *i.e.*, the current node — therefore it only requires a small amount of memory.
- **Cons** : Due to its greedy nature, hill climbing search can be easily trapped by local optima or flat state landscapes.



Local search

- Hill-climbing with sideways move: allow search to continue on a shoulder or “flat” local optima.
- Stochastic hill climbing: randomly choose a successor with better evaluation values
- First-choice hill climbing: choose the first successor with better evaluation value
- Random restart hill climbing: restart with a randomly generated initial state until success

Is hill climbing algorithm useful?

- The success of hill climbing depends very much on the shape of the state-space landscape:
  - if there are few local maxima and plateaux, random-restart hill climbing will find a good solution very quickly.
- On the other hand, many real problems have a landscape that looks more like a widely scattered family of balding porcupines on a flat floor, with miniature porcupines living on the tip of each porcupine needle, *ad infinitum*.
  - **Despite this, a reasonably good local maximum can often be found after a small number of restarts.**
- Can you think of how to systematically improve hill-climbing search?

Is hill climbing algorithm useful?

- The success of hill climbing depends very much on the shape of the state-space landscape:
  - if there are few local maxima and plateaux, random-restart hill climbing will find a good solution very quickly.
- On the other hand, many real problems have a landscape that looks more like a widely scattered family of balding porcupines on a flat floor, with miniature porcupines living on the tip of each porcupine needle, *ad infinitum*.
  - **Despite this, a reasonably good local maximum can often be found after a small number of restarts.**
- Can you think of how to systematically improve hill-climbing search?

Simulated annealing

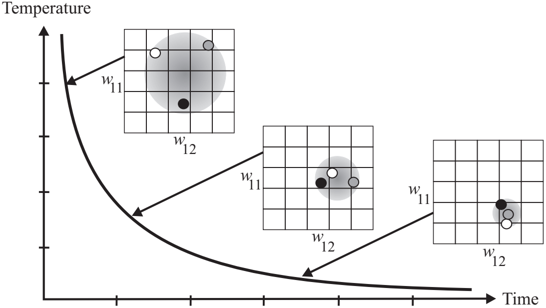
- A hill-climbing algorithm that never makes "downhill" moves toward states with lower value (or higher cost) is guaranteed to be **incomplete**, because it can get stuck on a local maximum.
- In contrast, a purely random walk — that is, moving to a successor chosen uniformly at random from the set of successors — is **complete** but **extremely inefficient**.
- Therefore, it seems reasonable to try to combine hill climbing with a random walk in some way that yields both efficiency and completeness, which motivates the **Simulated Annealing (SA)** method.

Simulated annealing

Annealing in metals

- Heat the solid state metal to a high temperature
- Cool it down very slowly according to a specific schedule.

If the heating temperature is sufficiently high to ensure random state and the cooling process is slow enough to ensure thermal equilibrium, then the atoms will place themselves in a pattern that corresponds to the global energy minimum of a perfect crystal.



Simulated annealing — shake the ping-pong ball

Shake the ping-pong ball

Imagine the task of getting a ping-pong ball into the deepest crevice in a bumpy surface.

- If we just let the ball roll, it will come to rest at a local minimum.
- If we shake the surface, we can bounce the ball out of the local minimum.
- The trick is to shake just hard enough to bounce the ball out of local minima but not hard enough to dislodge it from the global minimum.

The simulated-annealing solution is to start by shaking hard (i.e., at a high temperature) and then gradually reduce the intensity of the shaking (i.e., lower the temperature).

Simulated annealing — shake the ping-pong ball

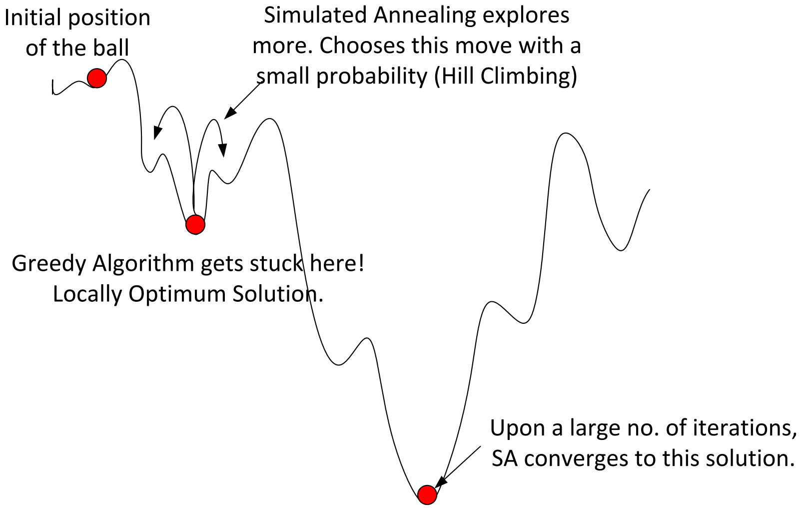
Shake the ping-pong ball

Imagine the task of getting a ping-pong ball into the deepest crevice in a bumpy surface.

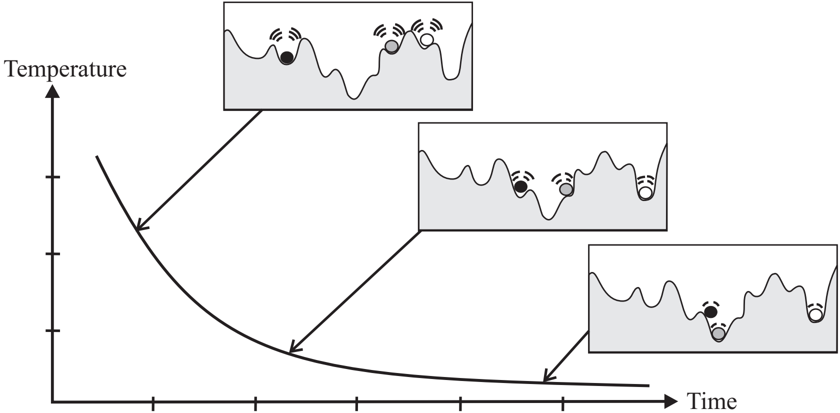
- If we just let the ball roll, it will come to rest at a local minimum.
- If we shake the surface, we can bounce the ball out of the local minimum.
- The trick is to shake just hard enough to bounce the ball out of local minima but not hard enough to dislodge it from the global minimum.

The simulated-annealing solution is to start by shaking hard (i.e., at a high temperature) and then gradually reduce the intensity of the shaking (i.e., lower the temperature).

Simulated annealing — shake the ping-pong ball



Simulated annealing — shake the ping-pong ball

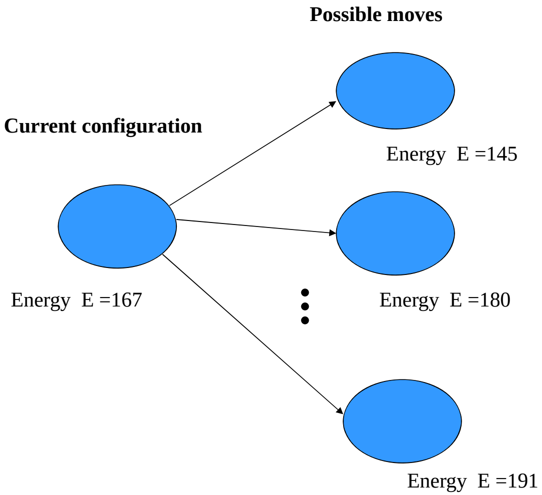


Simulated annealing search

Simulated annealing search

- Initialize *current* to starting state
- **For** $i = 1$ to ∞
  - **If** T(i) $= 0$ **return** *current*
  - Let *next* $=$ random successor of *current*
  - Let $\Delta =$ value(*next*) $-$ value(*current*)
  - **If** $\Delta > 0$ **then** let *current* $=$ *next*
  - **Else** let *current* $=$ *next* with probability $exp(\Delta / T(i))$

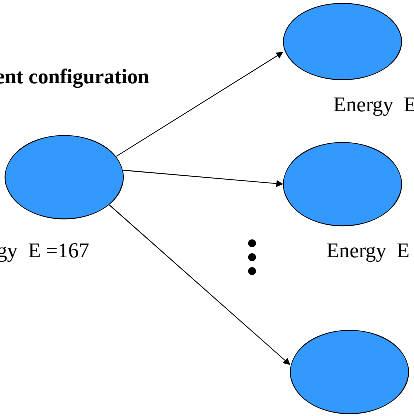
Simulated annealing search



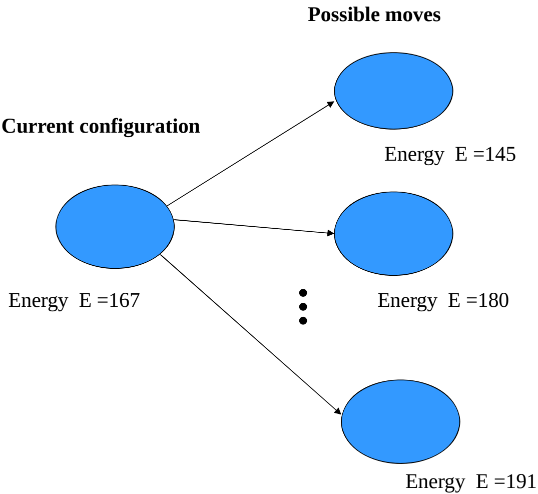
Simulated annealing search

$$\Delta E = E_{NEXT} - E_{CURRENT}$$
$$= 145 - 167 = -22$$
$$p(Accept) = e^{\Delta E / T} = e^{-22/T}$$

Sometimes accept!



Simulated annealing search



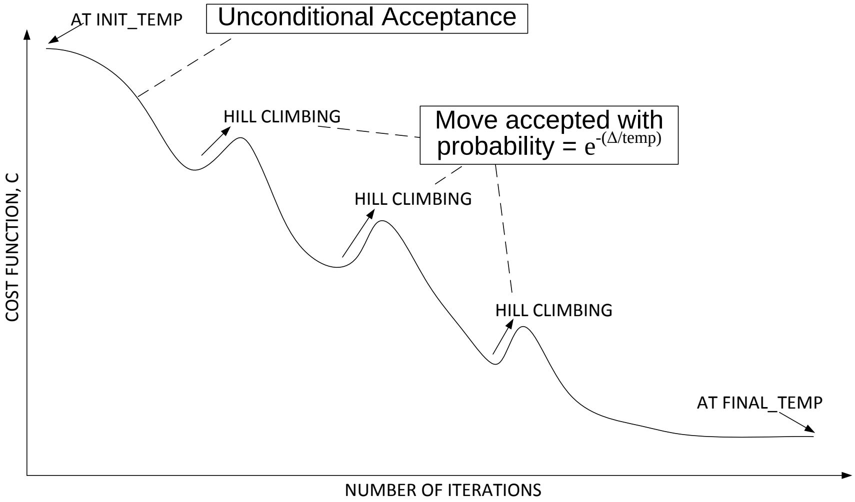
$$\Delta E = E_{NEXT} - E_{CURRENT}$$
$$= 180 - 167 > 0$$
$$p(Accept) = 1$$

Always accept!

Animation: simulated annealing

http://en.wikipedia.org/wiki/File:Hill_Climbing_with_Simulated_Annealing.gif

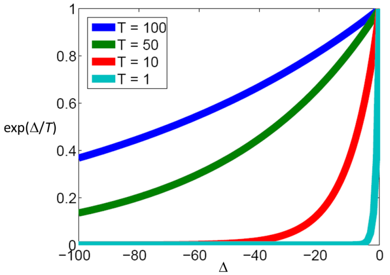
Convergence of simulated annealing



Effect of temperature

The probability decreases as the "temperature" T goes down.

- "Bad" moves are more likely to be allowed at the start when T is high,
- and they become more unlikely as T decreases.

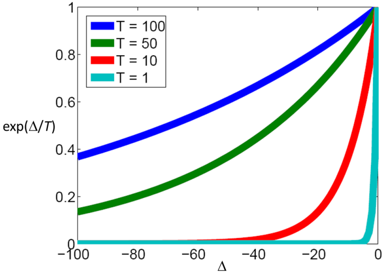


- One can prove: If temperature decreases slowly enough, then simulated annealing search will find a global optimum with probability approaching one
- However:
  - This usually takes impractically long
  - The more downhill steps you need to escape a local optimum, the less likely you are to make all of them in a row
  - **More modern techniques: general family of Markov Chain Monte Carlo (MCMC) algorithms for exploring complicated state spaces**

Effect of temperature

The probability decreases as the "temperature" T goes down.

- "Bad" moves are more likely to be allowed at the start when T is high,
- and they become more unlikely as T decreases.

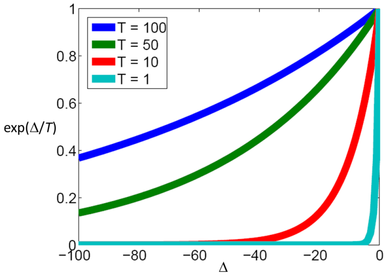


- One can prove: If temperature decreases slowly enough, then simulated annealing search will find a global optimum with probability approaching one
- However:
  - This usually takes impractically long
  - The more downhill steps you need to escape a local optimum, the less likely you are to make all of them in a row
  - More modern techniques: general family of Markov Chain Monte Carlo (MCMC) algorithms for exploring complicated state spaces

Effect of temperature

The probability decreases as the "temperature" T goes down.

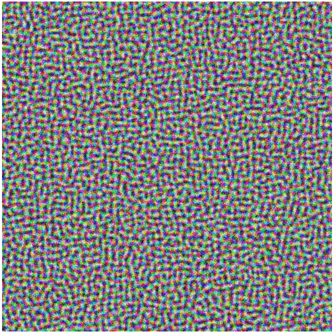
- "Bad" moves are more likely to be allowed at the start when T is high,
- and they become more unlikely as T decreases.



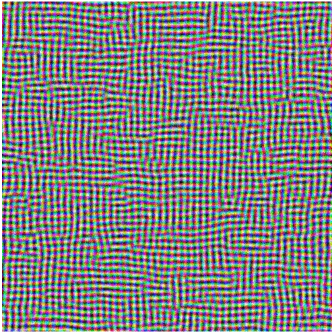
- One can prove: If temperature decreases slowly enough, then simulated annealing search will find a global optimum with probability approaching one
- However:
  - This usually takes impractically long
  - The more downhill steps you need to escape a local optimum, the less likely you are to make all of them in a row
  - **More modern techniques: general family of Markov Chain Monte Carlo (MCMC) algorithms for exploring complicated state spaces**

Effect of time cooling schedule

Example illustrating the effect of cooling schedule on the performance of simulated annealing. The problem is to rearrange the pixels of an image so as to minimize a certain potential energy function, which causes similar colors to attract at short range and repel at a slightly larger distance. The elementary moves swap two adjacent pixels. These images were obtained with a fast cooling schedule (left) and a slow cooling schedule (right), producing results similar to amorphous and crystalline solids, respectively.



Simulated Annealing Fast



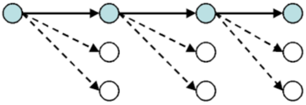
Simulated Annealing Slow

Solve the traveling salesman (TSM) problem by simulated annealing

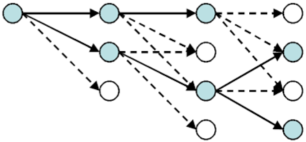
http://www.youtube.com/watch?v=SC5CX8drAtU

Local beam search

- Start with k randomly generated states
- At each iteration, all the successors of all k states are generated
- If any one is a goal state, stop; else select the k best successors from the complete list and repeat



Greedy search

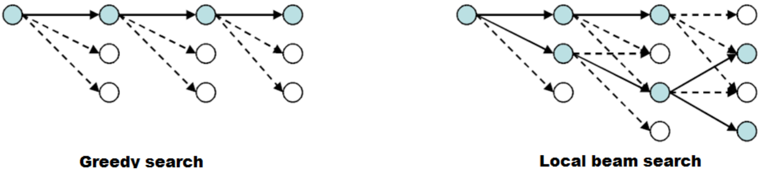


Local beam search

Is this the same as running k greedy searches in parallel?

Local beam search

- Start with k randomly generated states
- At each iteration, all the successors of all k states are generated
- If any one is a goal state, stop; else select the k best successors from the complete list and repeat

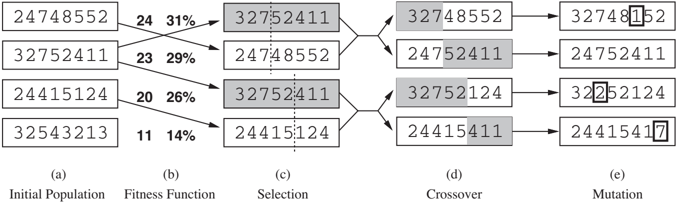


Is this the same as running k greedy searches in parallel?

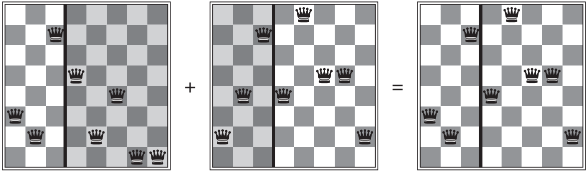
- In a random-restart search, each search process runs independently of the others.
- In a local beam search, useful information is passed among the parallel search threads.

Genetic search

The genetic algorithm, illustrated for digit strings representing 8-queens states. The initial population in (a) is ranked by the fitness function in (b), resulting in pairs for mating in (c). They produce offspring in (d), which are subject to mutation in (e).



The 8-queens states corresponding to the first two parents in figure (c) above and the first offspring in figure (d) above. The shaded columns are lost in the crossover step and the unshaded columns are retained.



Copyright statement