CS 440: Introduction to Artificial Intelligence

Lecture 11

Matthew Stone

October 14, 2015

Recap— Search

- Initial state
- Possible actions in each state
- Transition model:
  Takes state and action and gives new state
- Goal test
  Describes whether state is what you want
- Path cost
  Says how easy or hard action sequence is

Recap— Search

Tips on formalizing search problems

- Represent states simply and precisely
- Make actions as simple and flexible as possible
- Count on search strategy to make good choices
- Analyze complexity and improve if necessary

A* Search

Practical variant of heuristic search

- Go-to-method for informed search
- Assumes you have a good way to measure progress

Basic idea: Explore the search node that looks most promising

- Measure progress by actual cost already incurred
- *plus* estimate of cost remaining

Thought Experiment

Compare

- One search problems where solution has depth d
- Two search problems where solution has depth $d/2$

Alternative Search Algorithm

Suppose you can construct a unique goal state.
Suppose you can apply actions backwards.

- Conduct two searches in parallel
- Search A starts from start, looks for goal forward
- Search B starts from goal, looks for start backward
- Solution happens when A and B meet

(Demo)

Limits of admissibility

- Lower bounds on solution can be weak
- A* explores too much of the search space
- May be better to give up optimality
- Settle for solution that's "good enough"

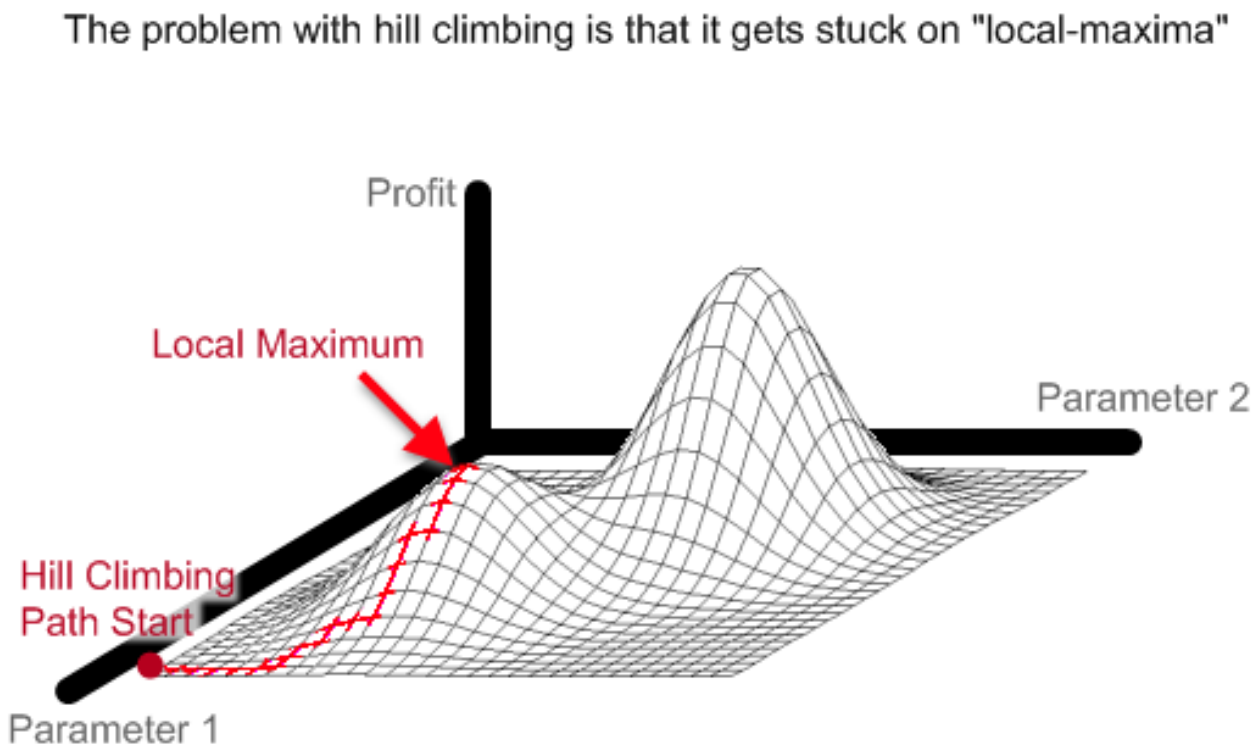
Strategy 1: Hill climbing

- Construct random costly solution, then loop:
- Construct children of current state
- If children are more costly return current
- Otherwise set current to best child and repeat

Assumptions

- Assumes it's easy to construct costly solution
- Assumes it's easy to modify solutions
- Also works with continuous spaces
- Sample or use analytic gradient

Hill Climbing weaknesses



from `http://www.maxdama.com/2008_07_01_archive.html`

Hill Climbing weaknesses

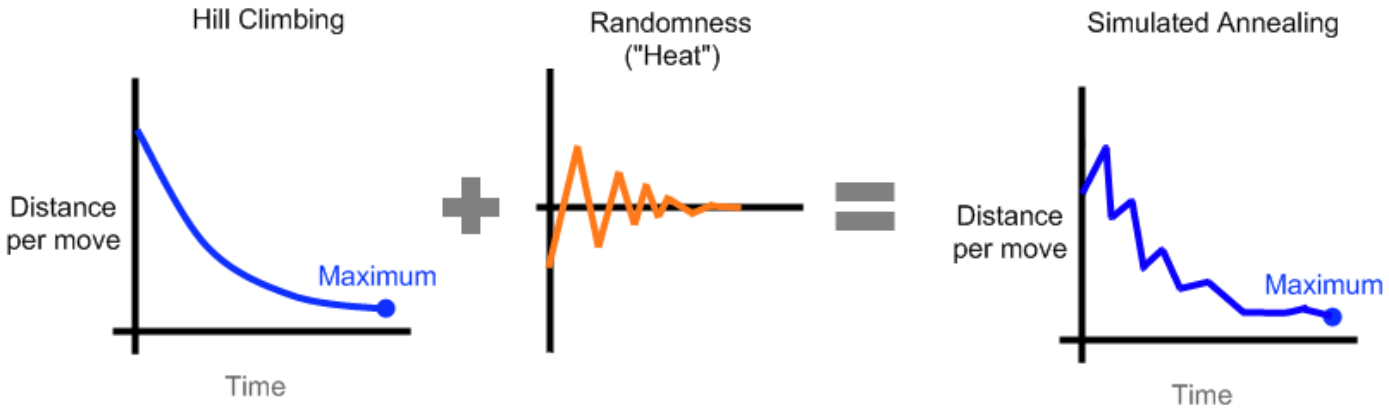
- Plateau problem
  Can't tell locally which way to go
  because all solutions look equally good
- Ridge problem
  Higher than surrounding areas but has slope upward
  Probes may not detect upward direction

see `http://wwwic.ndsu.edu/juell/vp/cs724s00/hill_climbing/index.html`

Strategy 2: Simulated annealing

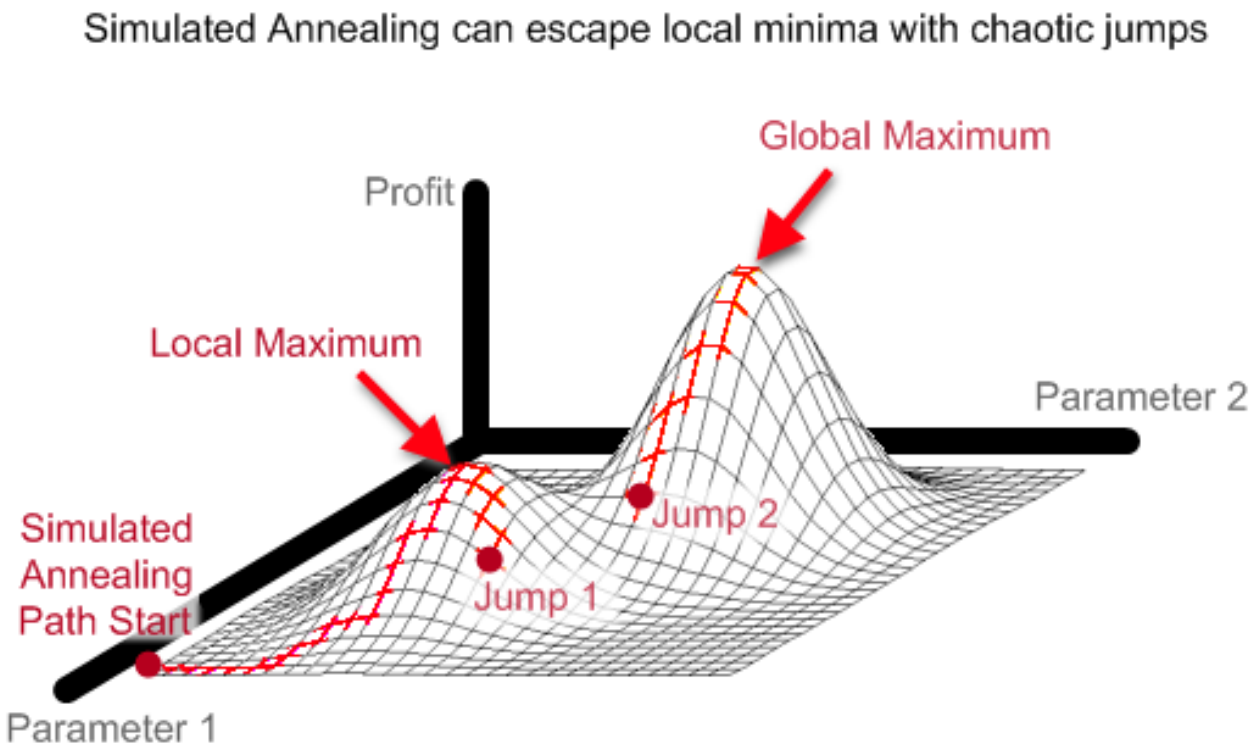
- Include a little randomness to avoid getting stuck
- More randomness early
  when you're likely to be further from good solution
- Mirrors process of metalwork
- Get metal in low-energy state by heating it up
  and cooling it slowly

Simulated annealing



from `http://www.maxdama.com/2008_07_01_archive.html`

Simulated annealing



from `http://www.maxdama.com/2008_07_01_archive.html`

Simulated annealing

- Construct random costly solution, then loop:
- Construct children of current state
- With probability t pick random child
- Otherwise pick least costly child
- Decrease t

Example search: Image synthesis

- Given raw image data
- Find representation in terms of polygons that approximates the image closely
- Demo at `http://alteredqualia.com/visualization/evolve/`

Strategy 3: Genetic Search

- Assumes it's easy to construct costly solution
- Assumes you can construct new solution from pieces of old ones
- Assumes there are many ways to change solution
- Assumes attributes of solutions are often largely independent

Genetic search

- Create population: set of costly solutions then loop:
- Rank population by fitness
- Create next generation
  - Pick two parents as a function of fitness
  - Construct child by mixing parents' features
  - Mutate child
- Repeat
- Return best solutions found after fixed time

Genetic Search Visualization

- http://www.glauserweb.ch/gentore.htm
- http://www.rennard.org/alife/english/gavgb.html