

CS 4100/5100: Foundations of AI

More About Search

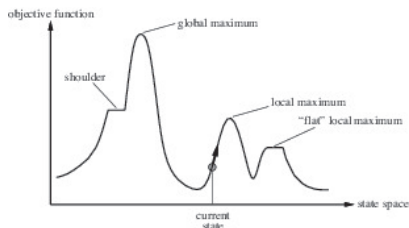
Instructor: Rob Platt
r.platt@neu.edu

College of Computer and information Science
Northeastern University

September 5, 2013

Optimization vs search

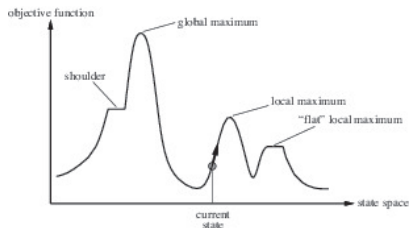
So far, "search" has translated to "tree search". But, suppose instead that the problem is to optimize some analytic objective function.



For example, suppose that you want to find the state where this function is maximized.

- ▶ notice that we might have a real-valued state space
- ▶ this is kind of like search in that we must find something. But, the objective function provides additional information...

Hill-climbing



Idea:

- ▶ evaluate direction of local steepest ascent. Move in that direction.
- ▶ notice that this strategy is applicable either to real-valued or discrete state spaces.
- ▶ for a discrete state space, this is the same as greedy best-first search if you take the optimization function to be the heuristic.

Hill-climbing Example

8-queens problem:

- ▶ find a configuration of 8 queens on a standard chess board such that no queen attacks another

Heuristic optimization fn:

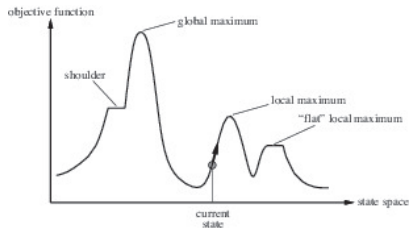
- ▶ # of pairs of queens that are attacking each other

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	♙	13	16	13	16
♙	14	17	15	♙	14	16	16
17	♙	16	18	15	♙	15	♙
18	14	♙	15	15	14	♙	16
14	14	13	17	12	14	12	18

Hill-climbing:

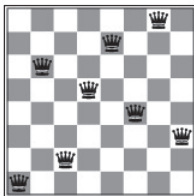
- ▶ numbers indicate the heuristic value of moving the queen in that column to that square
- ▶ move the queen that lowers the heuristic optimization function the most.

Problems w/ hillclimbing:



- ▶ It gets stuck in local minima.
- ▶ Difficult to negotiate "ridges" or "plateaux".

Problems w/ hillclimbing:



- ▶ It gets stuck in local minima.
- ▶ Difficult to negotiate "ridges" or "plateaux".

Possible solutions:

- ▶ random restarts
- ▶ allow a limited number of "sideways" moves

Simulated Annealing

Potential solution to the problems of hill-climbing:

Imagine a ball rolling around on the search surface (now we're trying to minimize, not maximize the fn).

- ▶ Ball naturally rolls into a local minimum.

However, now we're going to "shake" the world to introduce some stochasticity into the hill climbing process.

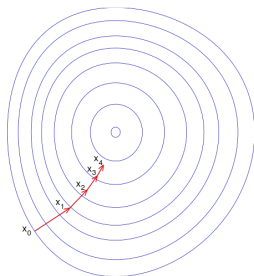
- ▶ Imagine shaking the world as the ball rolls over the search surface.
- ▶ "annealing" is a reference to metallurgy...

Simulated Annealing

How does this work exactly?

- ▶ Instead of moving in the direction of steepest descent on each step, we're going to pick a direction to move randomly.
- ▶ "Accept" the randomly chosen motion w/ probability proportional to how close the move is to the steepest descent direction.
 - ▶ probability of accepting the move decreases exponentially w/ the "distance" of the move from the direction of steepest descent.
 - ▶ for example, imagine a Gaussian distribution over move directions in a real-valued search space.
- ▶ Amount of "shaking" decreases as time goes on.
 - ▶ Probability of acceptance for non-steepest directions approaches zero as time goes on.
 - ▶ *i.e.* the constant in front of the exponent increases as time goes on...

Gradient descent



The same as hill-climbing except for continuous spaces.

1. calculate gradient
2. move a small distance in that direction.
3. goto step 1

Gradient descent

Suppose that we are descending the following objective function:

$$f(x, y, u, v) = (x - a_x)^2 + (y - a_y)^2 + (x - b_x)^2 + (y - b_y)^2.$$

What's this function look like?

What's the gradient?

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)^T.$$

where

$$\frac{\partial f}{\partial x} = 2(x - a_x) + 2(x - b_x), \quad \frac{\partial f}{\partial y} = 2(y - a_y) + 2(y - b_y)$$

Gradient descent

Simple update rule:

$$\mathbf{x} \leftarrow \mathbf{x} + \alpha \nabla_{\mathbf{x}} f(\mathbf{x}),$$

where

$$\mathbf{x} = (x, y)^T.$$

and α is a "small" step size.

Line search...

Gradient descent

Better update rule: **Newton-Raphson**

$$\mathbf{x} \leftarrow \mathbf{x} - \alpha H(\mathbf{x})^{-1} \nabla f(\mathbf{x}),$$

where H denotes the Hessian of f .

- ▶ what's a Hessian?

Essentially, Newton-Raphson fits a paraboloid to the function at \mathbf{x} and moves to the minimum of that paraboloid.

There are more efficient algorithms for various classes of convex optimization problems (linear programming, quadratic programming, *etc.*).

Search w/ non-deterministic actions

Up until now, our tree-search methods have assume deterministic actions.

- ▶ what are deterministic actions?

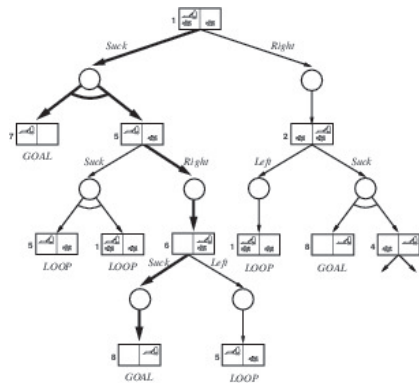
What if actions have stochastic outcomes?

- ▶ then we need to plan for multiple outcomes.
- ▶ this is achieved using an *and-or* search tree.

And-or search trees

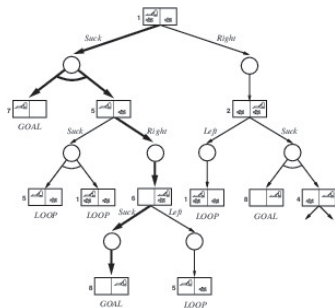
The "erratic-vacuum" world:

- ▶ when applied to a dirty square, the "suck" action cleans the square and sometimes cleans the adjacent square.
- ▶ when applied to a clean square, "suck" sometimes deposits dirt.



And-or search trees

The notion of a "plan" now becomes a "contingency plan" – a strategy for acting depending upon future observations.



- ▶ which are the "or" nodes?
- ▶ which are the "and" nodes?
- ▶ how do you create contingency plans?

And-or search algorithm

Depth first search version of the and-or search:

- ▶ recursive implementation of DFS
- ▶ when algorithm reaches an OR node: success if ANY one of the actions results in a successful path
- ▶ when algorithm reaches an AND node: success only if ALL of possible outcomes has a corresponding successful path.