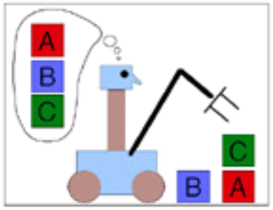
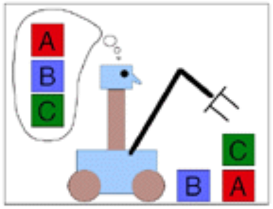


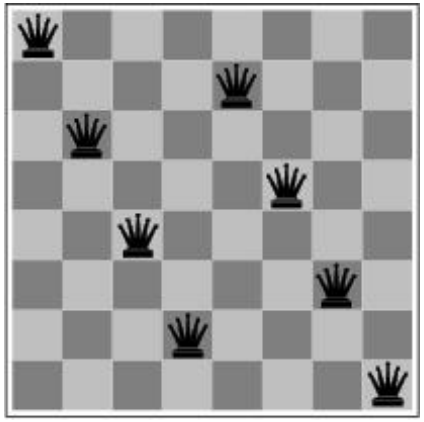
Optimization Problems and Local Search

Russell and Norvig 4.1

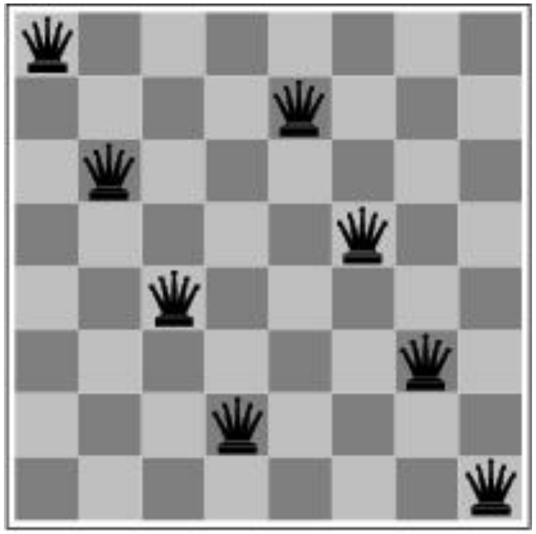
Optimization Problems



- Previously: systematic exploration of search space.
  - Path to goal is the solution
- For some problems path is irrelevant.
  - Example: 8-queens



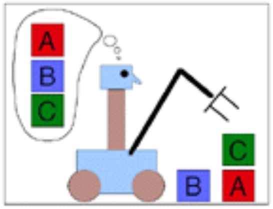
8 Queens



Stated as an optimization problem:

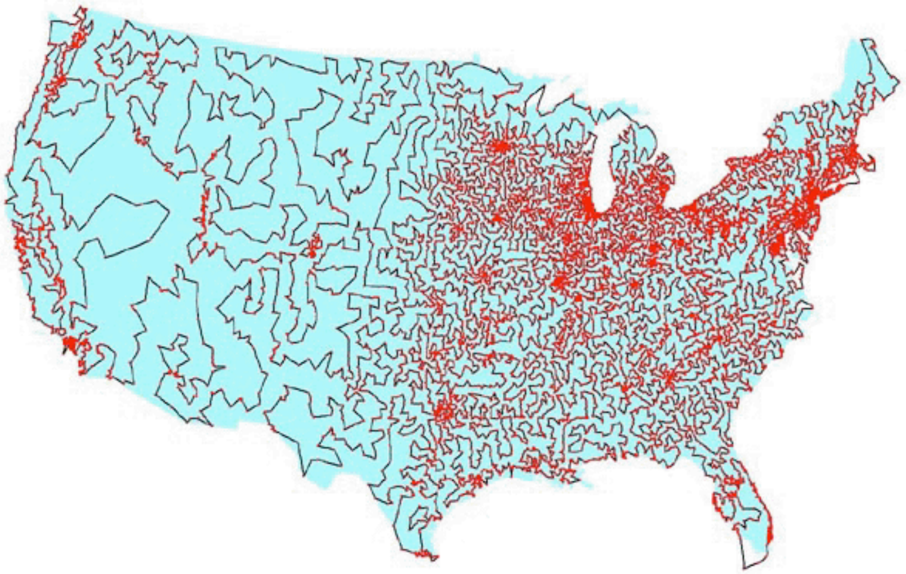
- State space: a board with 8 queens on it
- Objective/cost function: Number of pairs of queens that are attacking each other (quality of the state).

The Traveling Salesman Problem (TSP)



TSP: Given a list of cities and their pairwise distances, find a shortest possible tour that visits each city exactly once.

An optimal TSP tour through Germany’s 15 largest cities (one out of 14!/2)



13,509 cities and towns in the US that have more than 500 residents

http://www.tsp.gatech.edu/

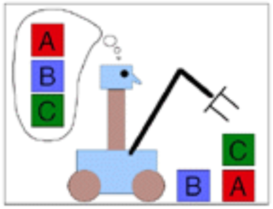
The Traveling Salesman Problem (TSP)

TSP: Given a list of cities and their pairwise distances, find a shortest possible tour that visits each city exactly once.

States?

Cost function?

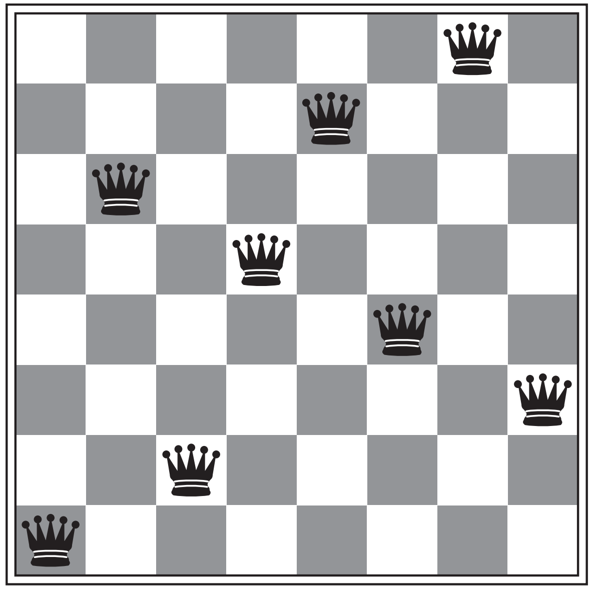
Local Search



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

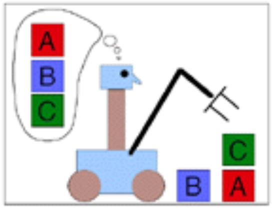
- Keep a current state, try to improve it by "locally" exploring the space of solutions
- Improve state by moving a queen to a position where fewer queens attack each other (neighboring state)
- Neighbors: move a queen in its column

Greedy local search



- Problem: can get stuck in a local minimum (happens 86% of the time for the 8-queens problem).

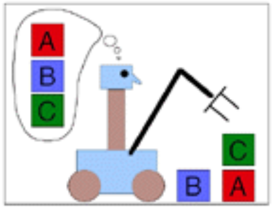
Local minima vs. local maxima



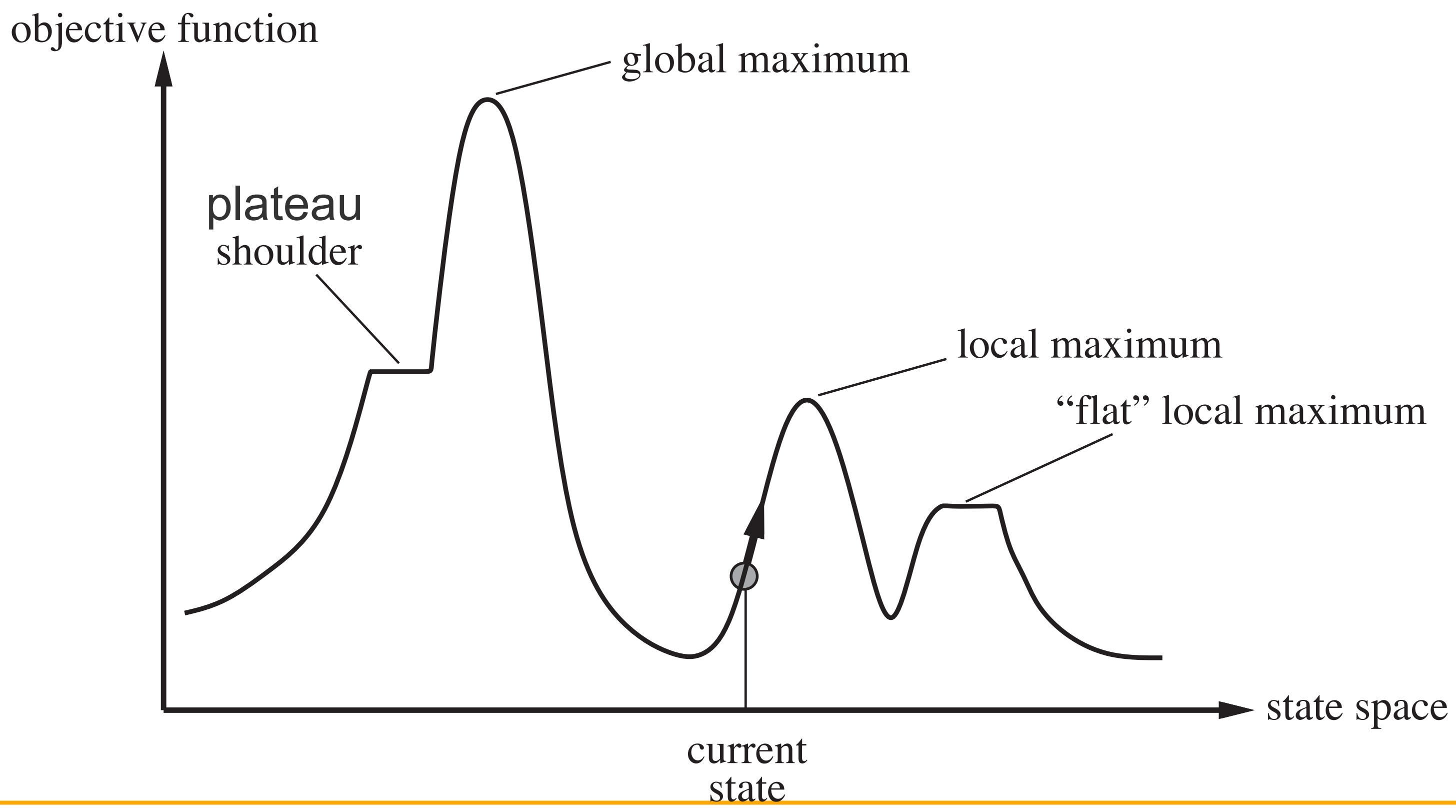
- Local search: find a local maximum or minimum of an objective function (cost function).

- local minima of a function $f(n)$ are the same of the maxima of $-f(n)$. Therefore, if we know how to solve one, we can solve the other.

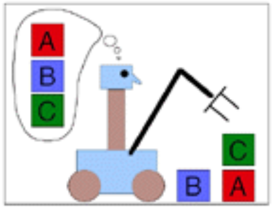
Hill-climbing



Try all neighbors and keep moves that improve the objective function the most



Hill-climbing



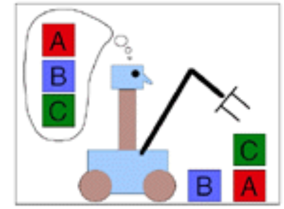
function HILL-CLIMBING(*problem*) **return** a state that is a local maximum

current ← MAKE-NODE(problem.INITIAL-STATE)
loop do
neighbor ← a highest valued successor of *current*
if neighbor.VALUE ≤ *current.*VALUE **then return** current.STATE
current ← *neighbor*

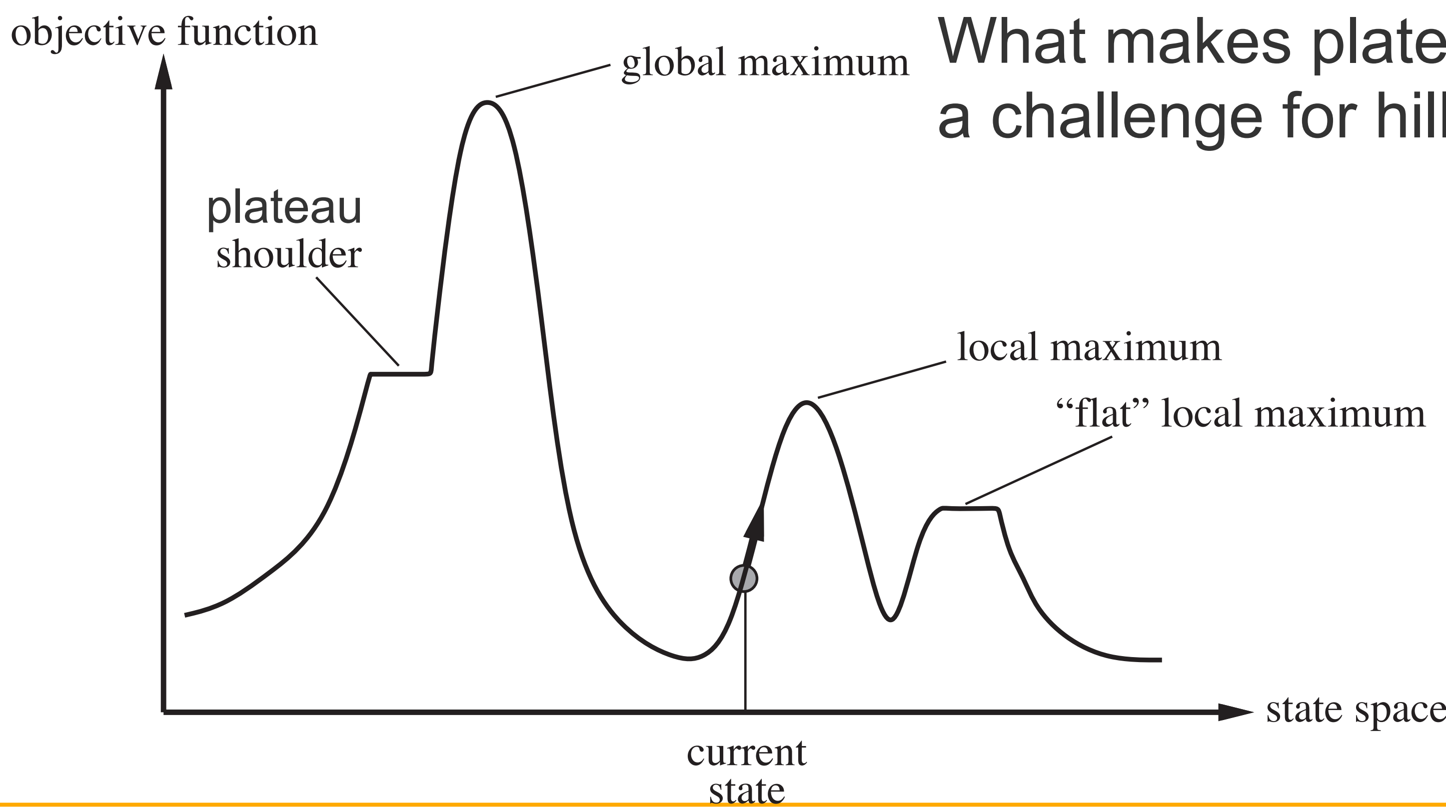
This flavor of hill-climbing is known as steepest ascent (steepest descent when the objective is minimization)

Finds local optimum 86% of the time for 8 queens problem.

Hill-climbing

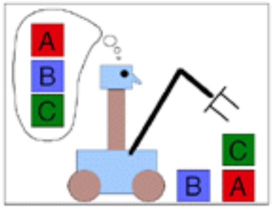


Try all neighbors and keep moves that improve the objective function the most



What makes plateaus a challenge for hill-climbing?

Formulating a problem as a local search problem



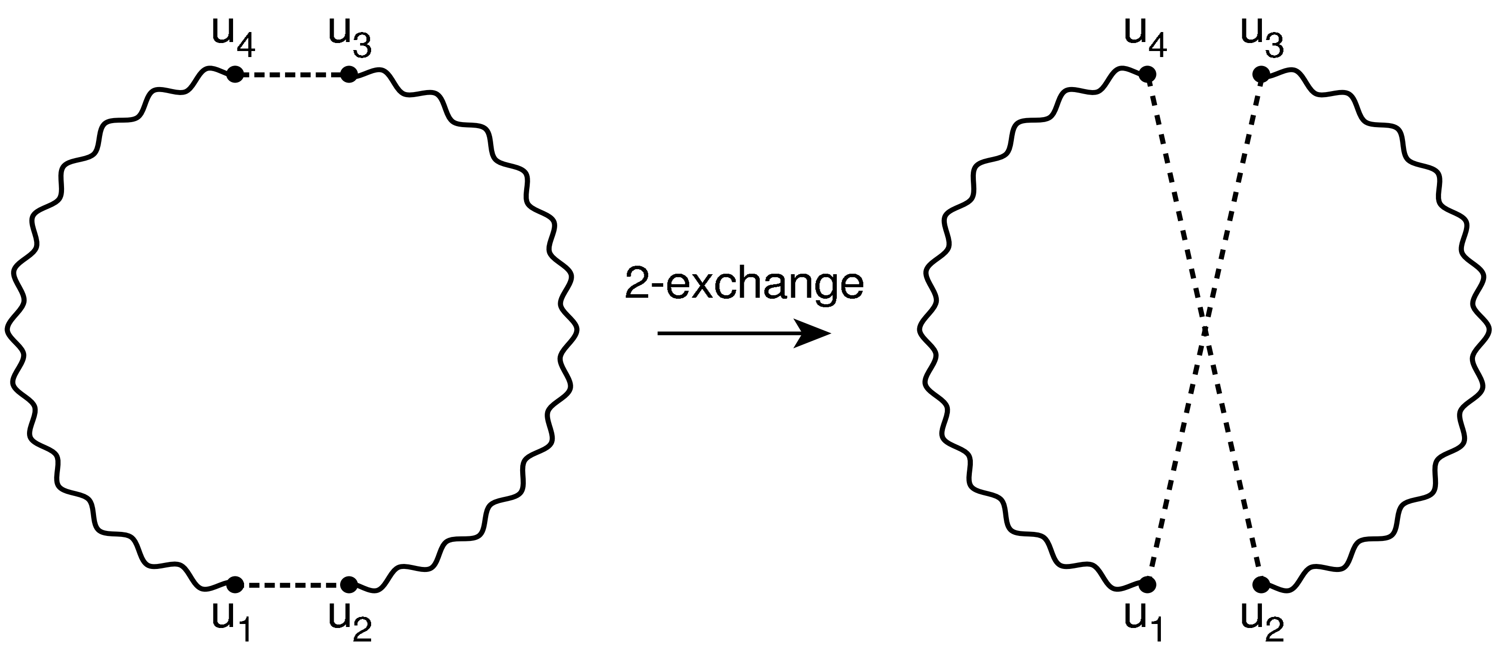
What you need to decide on:

- The possible states and their representation
- Choice of initial state
- Choice of neighborhood of a state
  - The neighborhood should be rich enough such that you don't get stuck in bad local optima
  - It should be small enough so that you can efficiently search the neighbors for the best local move

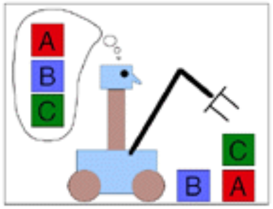
Solving TSP

Need to design a neighborhood that yields valid tours

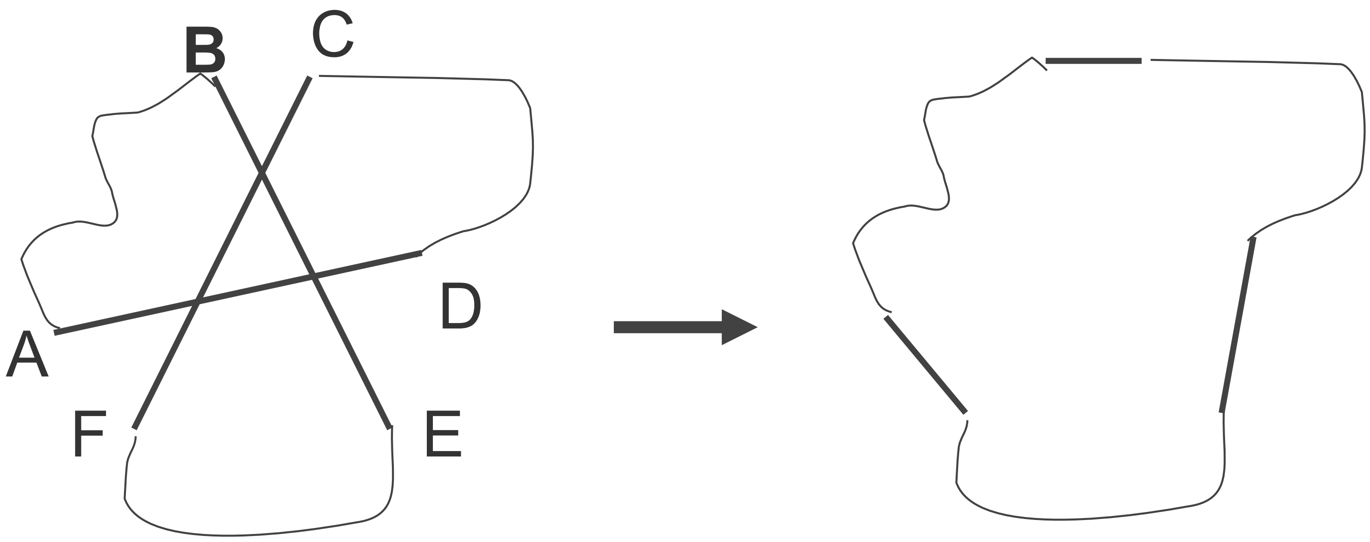
A 2-opt move:



3-opt

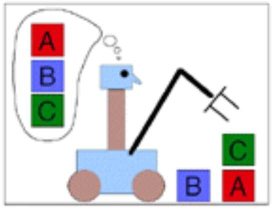


- Choose three edges from tour
- Remove them, and combine the three parts to a tour in the cheapest way to link them



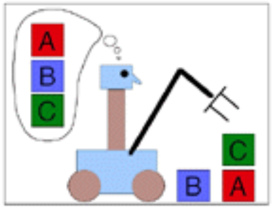
Source: University of Utrecht, www.cs.uu.nl/docs/vakken/na/na2-2005.ppt

Solving TSP (cont.)



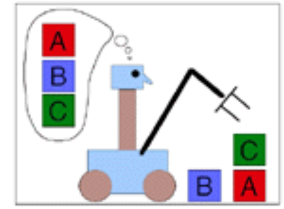
- 3-opt moves lead to better local minima than 2-opt moves.
- The Lin-Kernighan algorithm (1973): a λ-opt move - constructs a successor that changes λ cities in a tour
- Often finds optimal solutions.
- The best algorithm for TSP until 1989.

Variations of hill climbing



- Steepest ascent: choose the neighbor with the largest increase in objective function.
- Stochastic hill-climbing
  - Random selection among the uphill moves.
  - The selection probability can vary with the steepness of the uphill move.
- First-choice hill-climbing
  - Stochastic hill climbing, generating successors randomly until a better one is found.
- Random-restart hill-climbing
  - Choose best among several hill-climbing runs, each from a different random initial state.

Random Restart



- Suppose that the probability of failure in a single try is P_f
- The probability of failure in k trials:

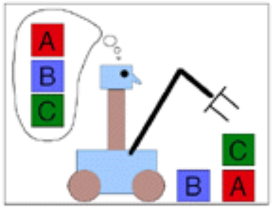
$$P_f(k\ trials) = (P_f)^k$$

$$P_s(k\ trials) = 1 - P_f(k\ trials) = 1 - (P_f)^k$$

- The probability of success can be made arbitrarily close to 1 by increasing k.
- Example: For the eight queens problem

$$P_s(100\ trials) = 0.9999997$$

Hill climbing for NP-complete problems



- NP-complete problems can have an exponential number of local minima.

- But:
  - Most instances might be easy to solve
  - Even if we can't find the optimal solution, a reasonably good local maximum can often be found after a small number of restarts.