Search in Complex Environments

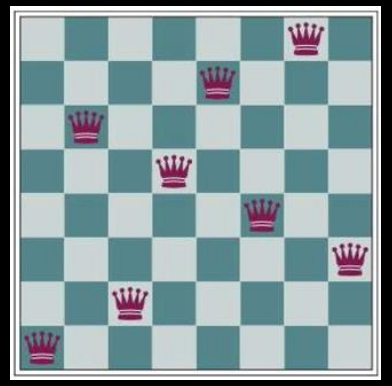
Asmaa Elbadrawy
PhD, Lecturer
IFT Program, ASU

Arizona State University

Given a problem,
find a solution state that satisfies certain conditions without worrying about the path to the solution.

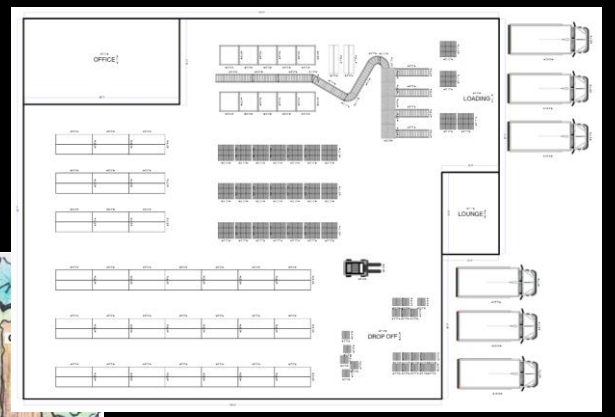
Example Problems

- 8-Queen

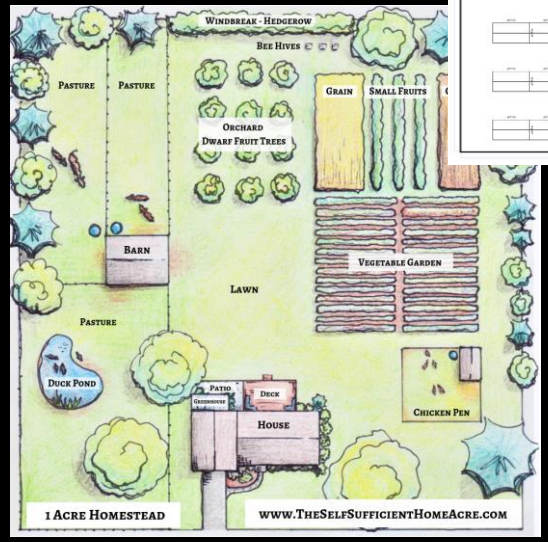


- Integrated Circuit Design

- Factory Floor Layout



- Crop Planning



Local Search & Optimization Algorithms

- Hill Climbing
- Simulated Annealing

Local Search

A search from an initial state to neighboring states. It does not keep track of paths nor previously-reached states. It just keeps exploring neighboring states until a solution is found.

Optimization Problem

It is a problem in which we try to find the best state according to some objective function. We say it is required to maximize the objective value.

Optimization Problem

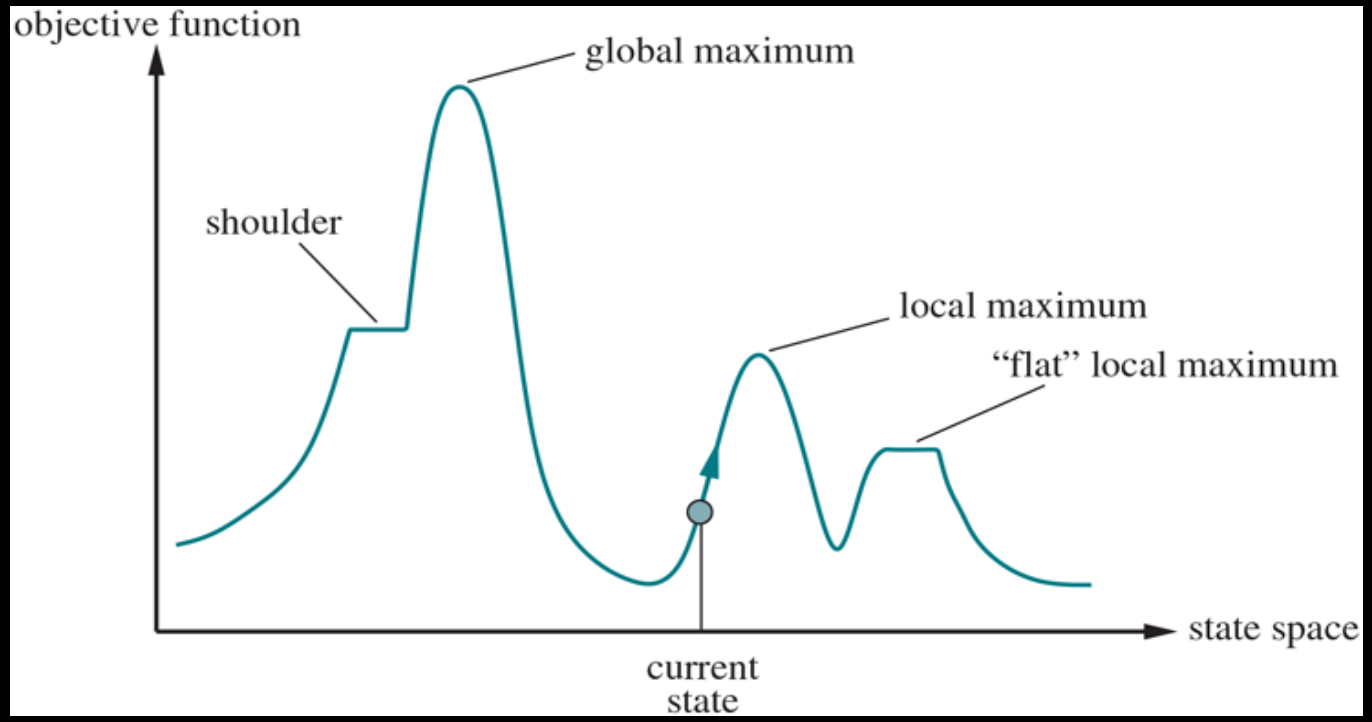


Fig 4.1, Russell & Norvig's Textbook

The objective value changes from one state to the next. Maximizing the objective is done by moving up-hill until a peak is found. This is referred to as Hill Climbing.

Hill Climbing starts from the initial state. With each step, it moves to the neighboring state with the highest value (called the direction of the steepest ascent). It terminates when a peak is reached. That is, there is no neighboring state that has a higher value. It does not necessarily terminate at a global maximum. It is not guaranteed to find the optimal solution (target state)!

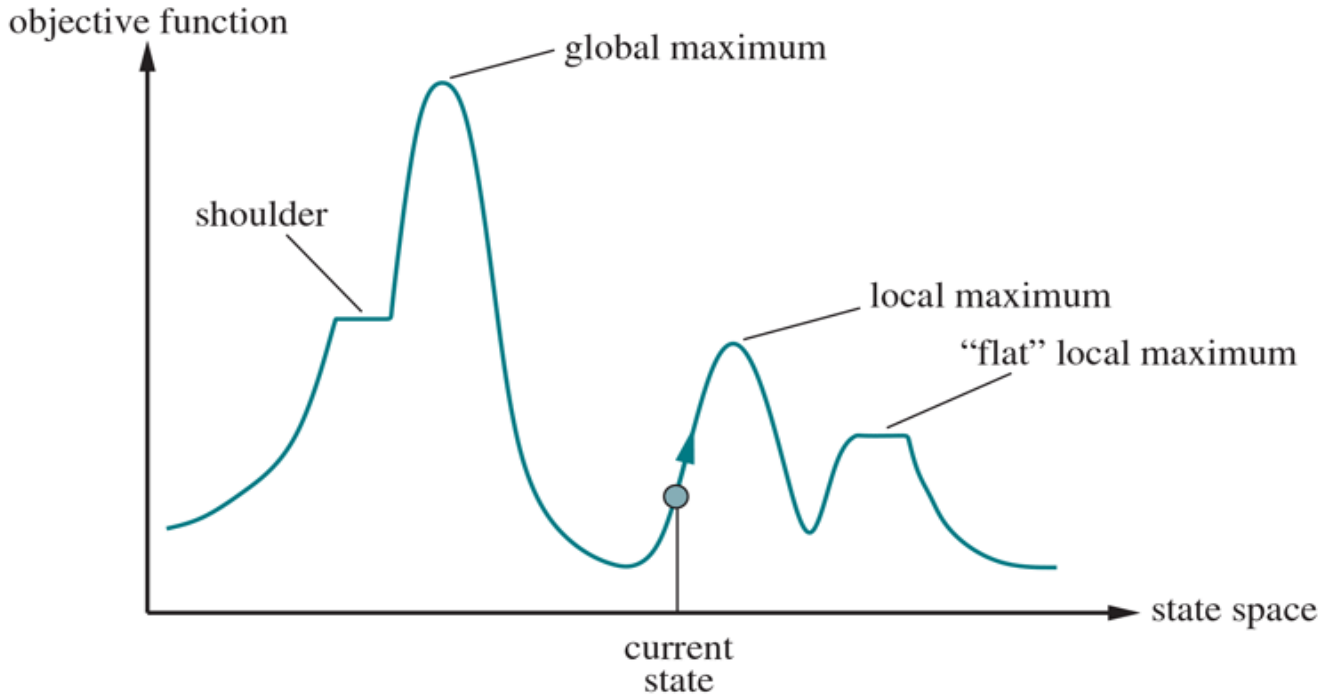


Fig 4.1, Russell & Norvig's Textbook

Hill climbing gets stuck at a local maximum or a plateau 86% of the time!
Success rate=14%
If it is allowed to continue on a plateau, success rate increases to 94%.

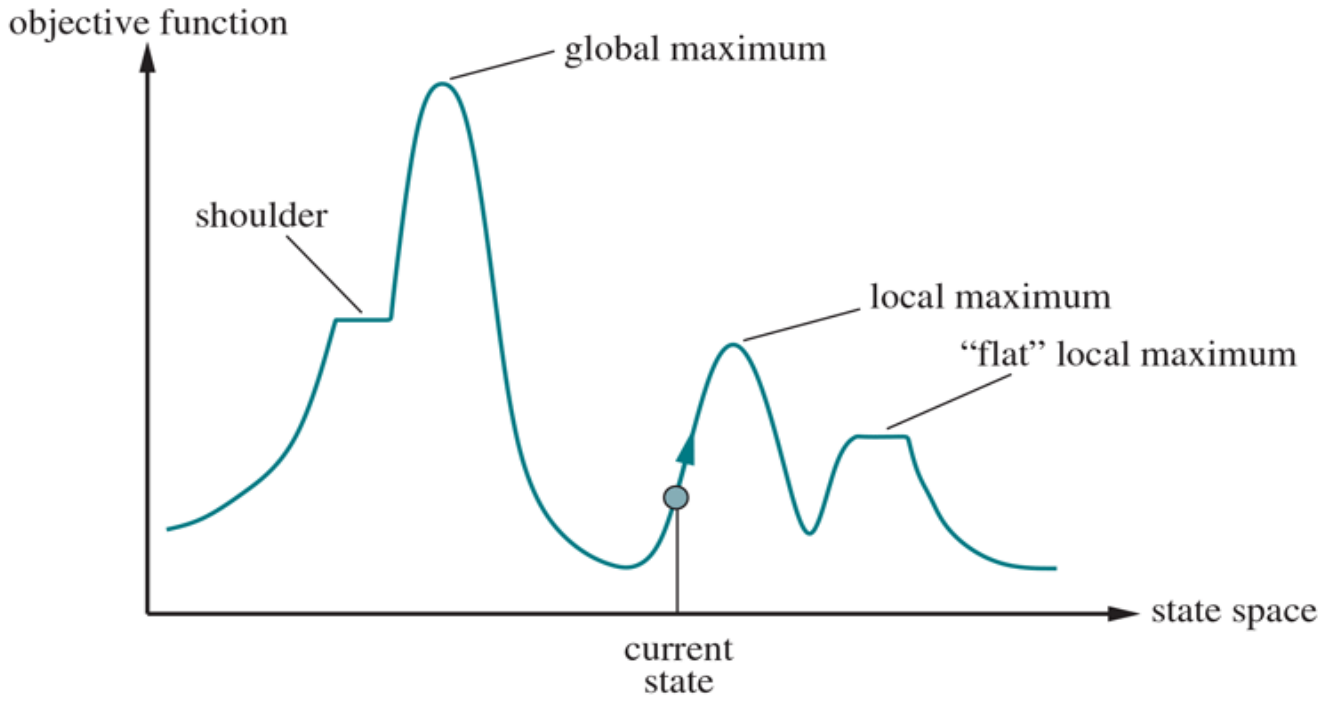


Fig 4.1, Russell & Norvig's Textbook

Another variation of Hill Climbing is Random-Restarting Hill Climbing. When the algorithm fails, it randomly generates a new initial state and restarts. On average, the algorithm restarts 7 times before it finds a solution.

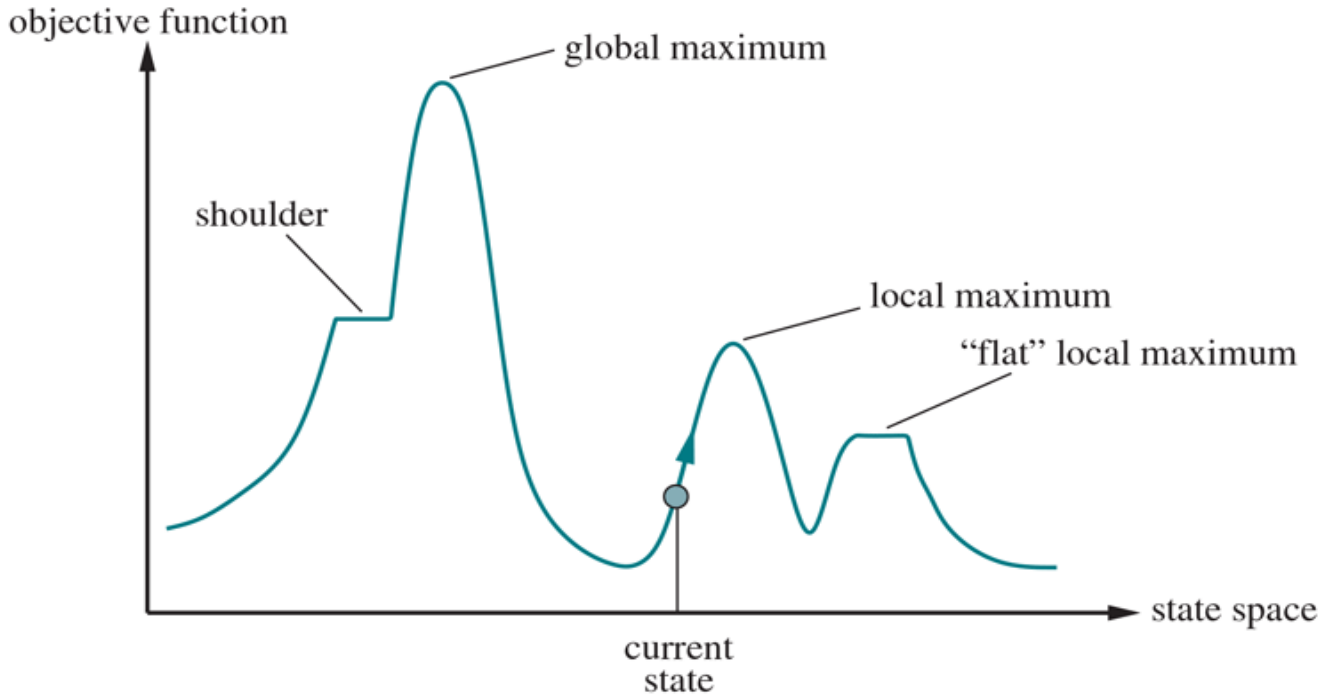


Fig 4.1, Russell & Norvig's Textbook

Prohibiting downhill moves leads to getting stuck at a local maximum (non-solution state).

Random walking will eventually reach a solution but is very inefficient.

Combining both strategies can lead to a more efficient and complete algorithm.

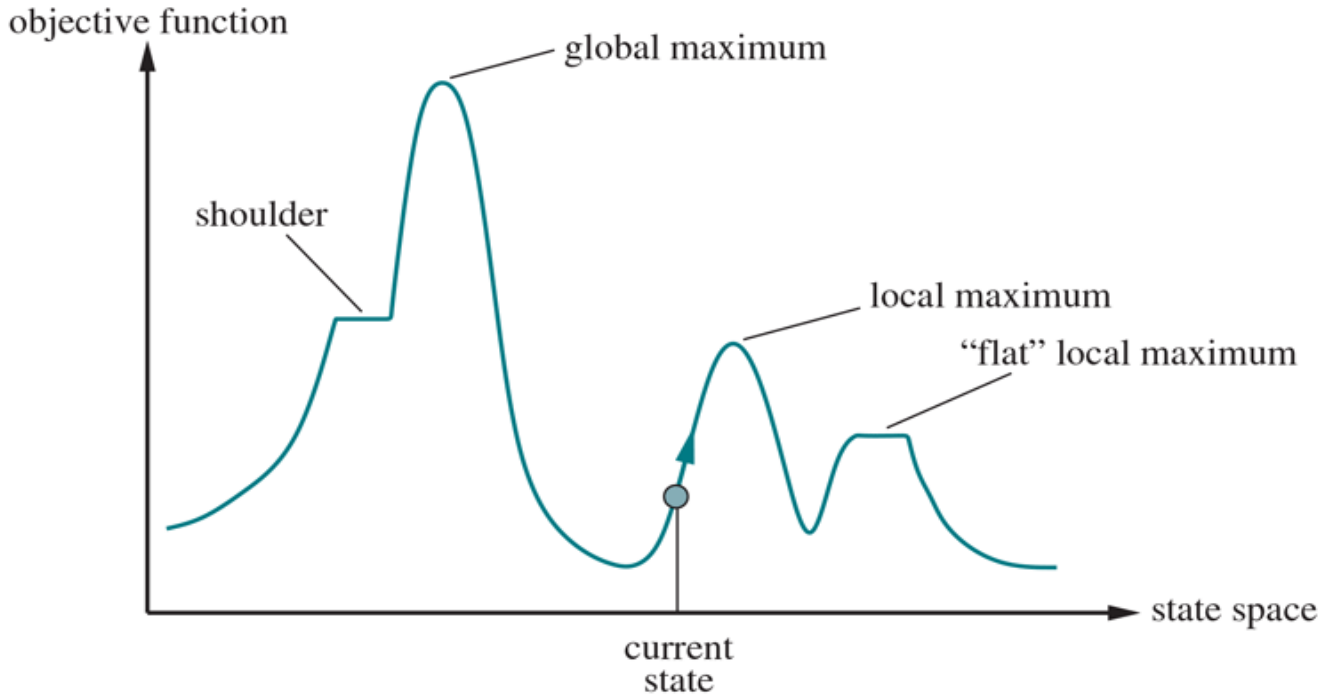


Fig 4.1, Russell & Norvig's Textbook

Simulated Annealing

Annealing is a process used to temper or harden metals and glass by heating them to high temperatures then gradually cooling them down.

Simulated Annealing works with ***cost***, not objective value. It tries to find the state with minimum cost. That is, it tries to find a ***global minimum*** point on the graph.

Simulated Annealing tries to “shake” the current state out of a local minimum to bounce it off towards, hopefully, a global minimum.

How is that done, mathematically?? By introducing controlled randomness

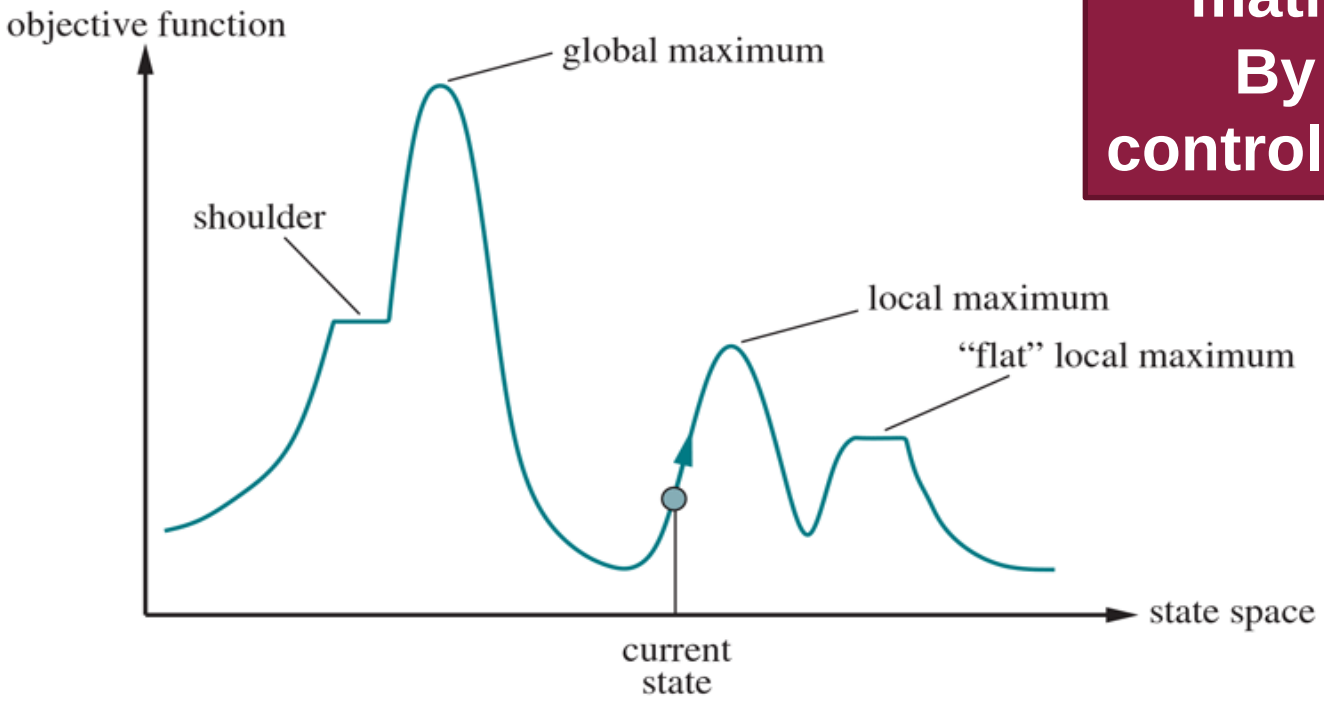


Fig 4.1, Russell & Norvig’s Textbook

Start at some random initial state

Repeat:

- Randomly generate a neighboring state

- If neighboring state is better that current state, accept
  Else, accept with some probability p, that decreases with T