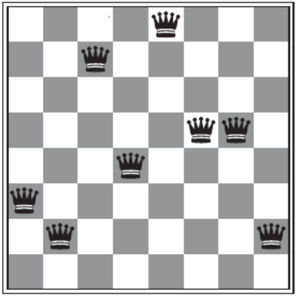
Optimization in discrete search space

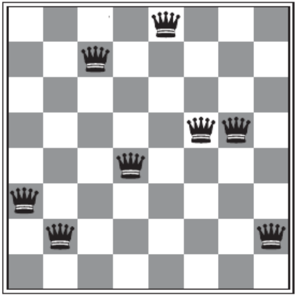
- Objective function
- Optimization over a discrete state space

Objective function: Cost vs. Fitness



3 2 7 4 8 5 5 2

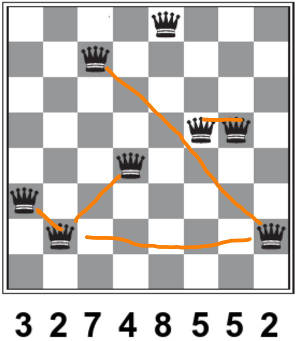
Objective function: Cost vs. Fitness



3 2 7 4 8 5 5 2

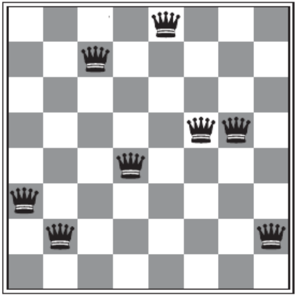
- State and State space

Objective function: Cost vs. Fitness



- State and State space
- Cost function $h = 5$

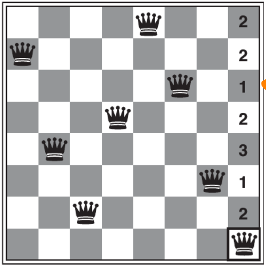
Objective function: Cost vs. Fitness



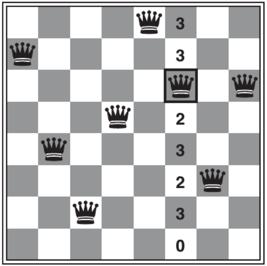
3 2 7 4 8 5 5 2

- State and State space
- Cost function $h = 5$
- Fitness function $= \binom{8}{2} - 5 = 23$

8 Queens Problem: 3 states

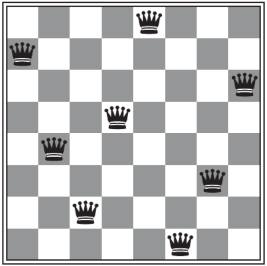


S1



S2

Soln.



S3

8 Queens Problem

- Total possible number of states?

$8 \times 8 \times \cdots 8$
$= 8^8$

8 Queens Problem

- Total possible number of states?
- How many neighbours does each state have?

$7 \times 8 = 56$

8 Queens Problem

- Total possible number of states?
- How many neighbours does each state have?
- Objective function?

Four search algorithms

- Hill climbing

Four search algorithms

- Hill climbing
- Simulated annealing

Four search algorithms

- Hill climbing
- Simulated annealing
- Local beam search

Four search algorithms

- Hill climbing
- Simulated annealing
- Local beam search
- Genetic algorithm

Steepest ascent Hill climbing algorithm

function HILL-CLIMBING(*problem*)

    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*

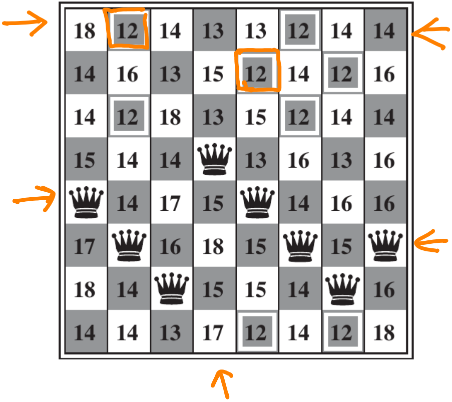
Steepest ascent Hill climbing algorithm

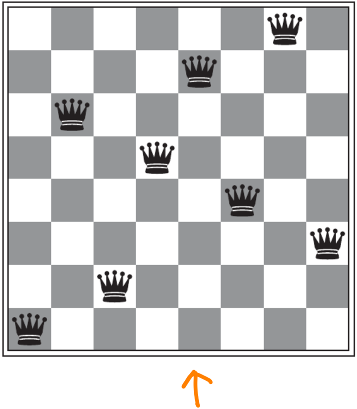
function HILL-CLIMBING(*problem*)

  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*

- Will this always work?

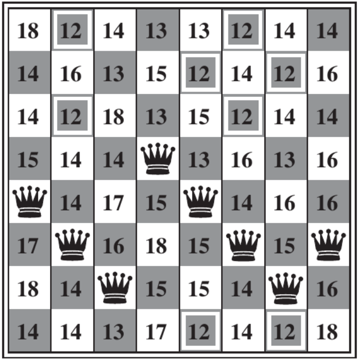
8-queens state

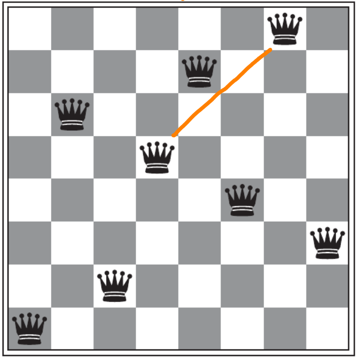




- 17 pairs of queens are in attacking position for the state on the left.

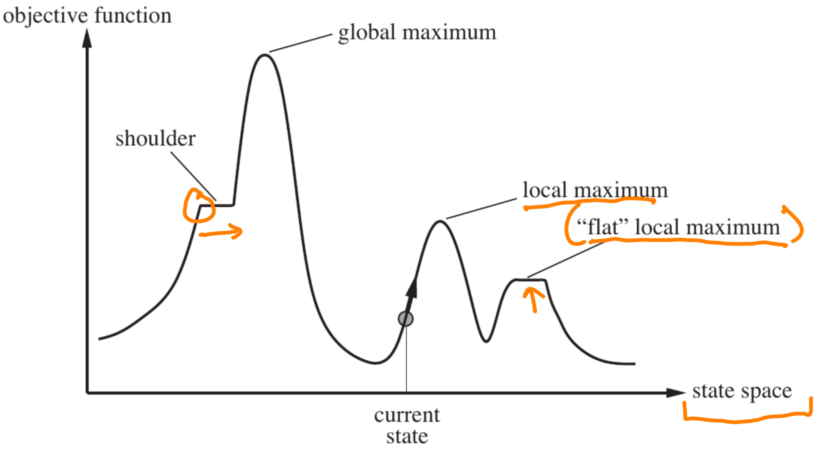
8-queens state





- 17 pairs of queens are in attacking position for the state on the left.
- After five steepest ascent steps, we reach a local maximum.

Landscape of the state-space



Possible solutions

- Sideways move

Possible solutions

- Sideways move
- N-consecutive sideways move

100 14% 94%

Possible solutions

- Sideways move
- N-consecutive sideways move
- Stochastic hill climbing

Possible solutions

- Sideways move
- N-consecutive sideways move
- Stochastic hill climbing
- Random-restart hill climbing

- Suppose, we have a coin that gives a head with probability p. Suppose we repeatedly toss the coin. What is the expected number of coin tosses before we get a heads?

$$E(x) = p \times 1 + (1-p)\left[E(x) + 1\right]$$

$$E(x) = \frac{1}{p}$$

Question

- Suppose, we have a coin that gives a head with probability p. Suppose we repeatedly toss the coin. What is the expected number of coin tosses before we get a heads?
- Suppose, steepest-ascent hill climbing succeeds in reaching the goal state with probability p. What is the expected number of starts required before the random-restart hill climbing will succeed?

$14\% = .14$

$\frac{1}{.14}$

Question

- Suppose, on average hill climbing succeeds in 4 steps and gets stuck (i.e. fails) in 3 steps. What is the expected number of **steps** before random-restart hill climbing will succeed? Assume probability of success to be .14.

$\frac{1}{.14}$

$$E(F+S) = \underbrace{E(F)} + \underbrace{E(S)}$$

$$= \underbrace{4} + \underbrace{\left(\frac{1}{.14} - 1\right) \times 3}$$

$\frac{1}{p} > 1$

$\approx \underbrace{22 \text{ steps}}$

Hill-climbing

- When will random-restart hill-climbing succeed in finding a good solution?
- Can we use it to solve the SAT problem? How can we define the state space?

$$\underbrace{(a \vee b \vee c)}_{2^3} \wedge \underbrace{(\neg e \vee f)}_{2^2} \leftarrow 2^5 = \underline{32}$$

$$7 \times 3 = \underline{21} \leftarrow$$