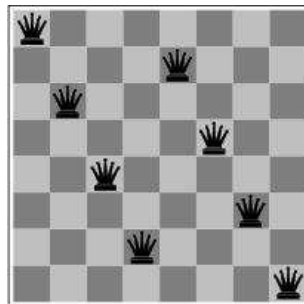# Local Search

*Note: this material was originated from the slides provided by Prof. **Padhraic Smyth***

Local search and optimization

- Previously: systematic exploration of search space.
  - Path to goal is solution to problem

- But, for some problems path is irrelevant.
  - E.g 8-queens

- Different algorithms can be used
  - Local search
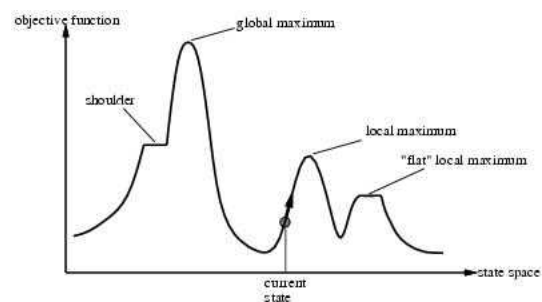   - ex) Hill-climbing

Local search and optimization

- Local search
  - Keep track of single current state
  - Move only to neighboring states
  - Ignore paths

- Advantages:
  - Use very little memory
  - Can often find reasonable solutions in large or infinite (continuous) state spaces.

- "Pure optimization" problems
  - Property 1: All states have an objective or heuristic function
  - Property 2: Goal is to find state with max (or min) objective value
  - Property 3: Does not quite fit into path-cost/goal-state formulation
  - Property 4: Local search can do quite well on these problems.

Hill-climbing search

- Very simple algorithm**:**
  - **continuously moves in the direction of increasing value**
  - terminates when a peak is reached
  - A kind of "greedy local best search"

- Value can be
  - Objective or heuristic function value

- Hill climbing does not look ahead of the immediate neighbors of the current state.

- Can randomly choose among the set of best successors, if multiple have the best value

- Characterized as "trying to find the top of Mount Everest while in a thick fog"
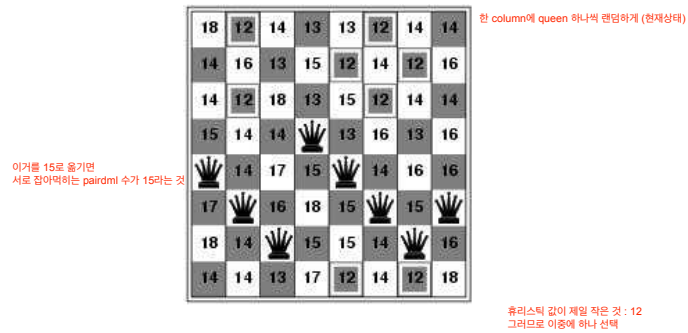
Hill climbing and local maxima

- When local maxima exist, hill climbing is suboptimal

- Simple (often effective) solution
  - Multiple random restarts

Hill-climbing example

- 8-queens problem, complete-state formulation
  - All 8 queens on the board in some configuration

- Successor function:
  - move a single queen to another square in the same column.

- Example of a heuristic function $h(n)$:
  - the number of pairs of queens that are attacking each other (directly or indirectly)
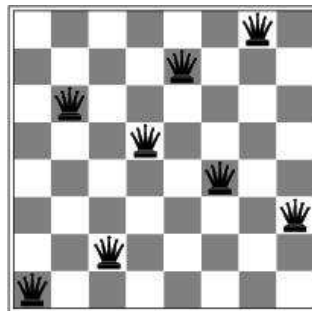 - (so we want to minimize this)

Hill-climbing example

한 column에 queen 하나씩 랜덤하게 (현재상태)

이거를 15로 옮기면
서로 잡아먹히는 pairdml 수가 15라는 것

휴리스틱 값이 제일 작은 것 : 12
그러므로 이중에 하나 선택

Current state: h=17

Shown is the h-value for each possible successor in each column

A local minimum for 8-queens

A local minimum in the 8-queens state space (h=1)

Performance of hill-climbing on 8-queens

- Randomly generated 8-queens starting states…

- 14% the time it solves the problem

- 86% of the time it get stuck at a local minimum

- However…
  - Takes only 4 steps on average when it succeeds
  - And 3 on average when it gets stuck
  - (for a state space with ~17 million states)

Hill-climbing with random restarts

- Different variations
  - Run a fixed number N of restarts
  - For each restart: run until termination

- Analysis
  - Say each search has probability p of success
    - E.g., for 8-queens, p = 0.14 of success 14%. 수치는 낮지만 여러번 해주면 99% 이상

  - Expected number n of restarts for over 99% success?
    Very Simple Interpretation (It may not be real !)
    $(0.86)^n < 0.01 \rightarrow n > \log_{0.86}0.01 = 30.53$ .30% 이상이면 99%의 확률로 찾을 수 있다
    독립사건
  - Expected maximum number of steps taken?
    4 X 31 = 124 (for a state space with ~17 million states)
    개의 state만 찾으면 99%의 확률로 찾을 수 있다