Local Search Algorithms

School of Electronic and Computer Engineering
Peking University

Wang Wenmin

Contents

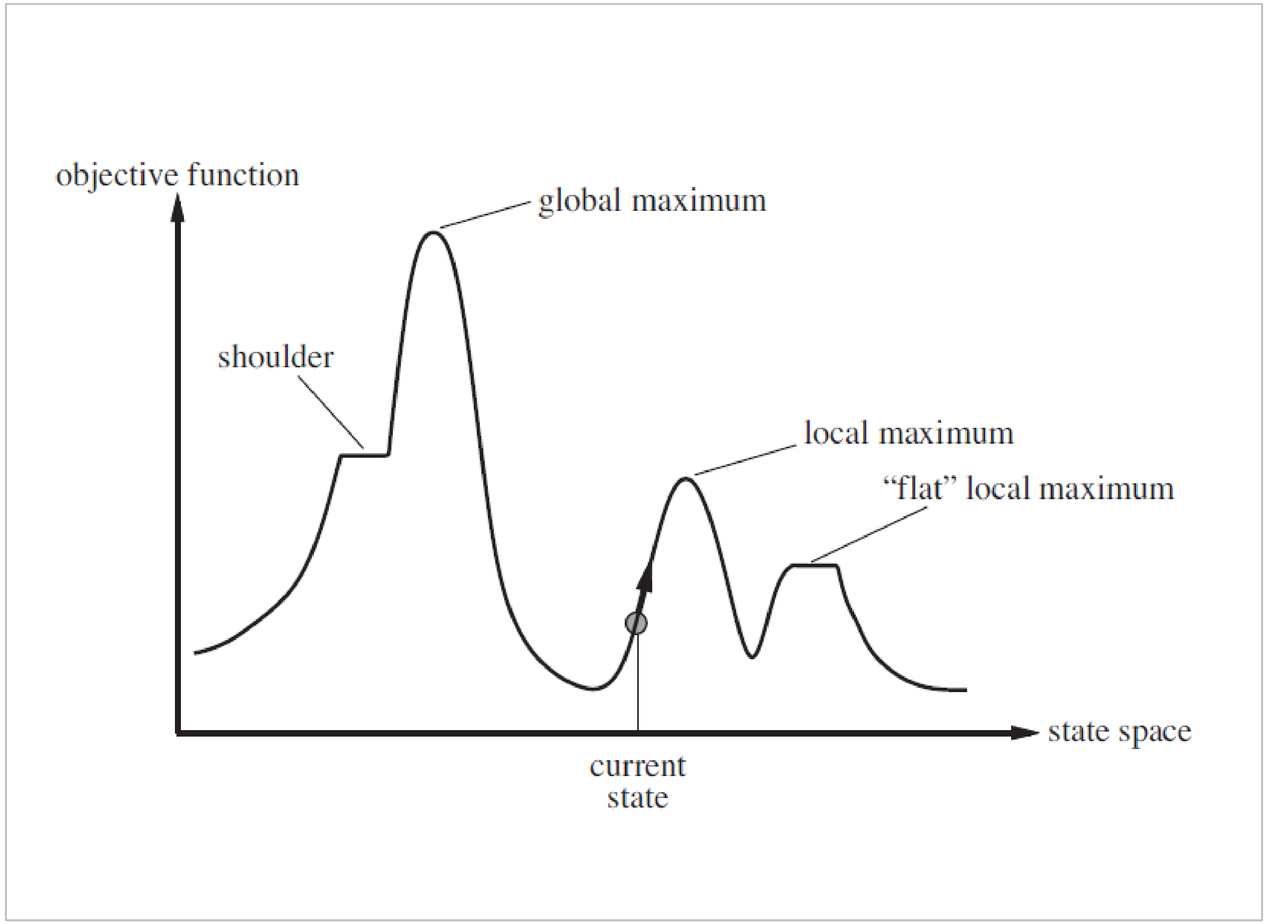
Hill-Climbing 爬山法

- A mathematical optimization technique which belongs to the family of local search.
  是一种属于局部搜索家族的数学优化方法。
- It is an iterative algorithm:
  - starts with an arbitrary solution to a problem,
  - then incrementally change a single element of the solution,
  - if it's a better solution, the change is made to the new solution,
  - repeating until no further improvements can be found.

  是一种迭代算法：开始时选择问题的一个任意解，然后递增地修改该解的一个元素，若得到一个更好的解，则将该修改作为新的解；重复直到无法找到进一步的改善。
- Most basic local search algorithm without maintaining a search tree.
  大多数基本的局部搜索算法都不保持一棵搜索树。

State-Space Landscape 状态空间地形图

- It can be explored by one of local search algorithms.
  可通过局部搜索算法对其进行搜索。
- A complete local search algorithm always finds a goal if one exists.
  一个完备的局部搜索算法总能找到一个存在的目标。
- an optimal local search algorithm always finds a global minimum or maximum.
  一个最优的局部搜索算法总能找到一个全局的最小或最大值。



Hill-Climbing Search Algorithm 爬山搜索算法

```
function HILL-CLIMBING(problem) returns a state that is a local maximum
    persistent: current, a node
                neighbor, a node
    current ← MAKE-NODE(problem.INITIAL-STATE)
    loop do
      neighbor ← a successor of current
      if neighbor.VALUE ≤ current.VALUE then return current.STATE
      current ← neighbor
```

Steepest-ascent version: at each step, current node is replaced by the best neighbor (the neighbor with highest value), else it reaches a “peak”.

最陡爬坡版：当前节点每一步都用最佳邻接点（具有最高值的邻接点）替换，否则到达一个“峰值”。

Hill-Climbing Search Algorithm 爬山搜索算法

- Hill-Climbing search algorithm is the most basic local search technique.
  爬山搜索算法是最基本的局部搜索方法。
- It often makes rapid progress toward a solution, because it is usually quite easy to improve a bad state.
  它常常会朝着一个解快速地进展，因为通常很容易改善一个不良状态。
- It is sometimes called greedy local search, because it grabs a good neighbor state without thinking ahead about where to go next.
  它往往被称为贪婪局部搜索，因为它只顾抓住一个好的邻接点的状态，而不提前思考下一步该去哪儿。
- Although greed is considered one of the “seven deadly sins”, it turns out that greedy algorithms often perform quite well.
  尽管贪婪被认为是“七宗罪” 之一，但是贪婪算法往往表现的相当好。

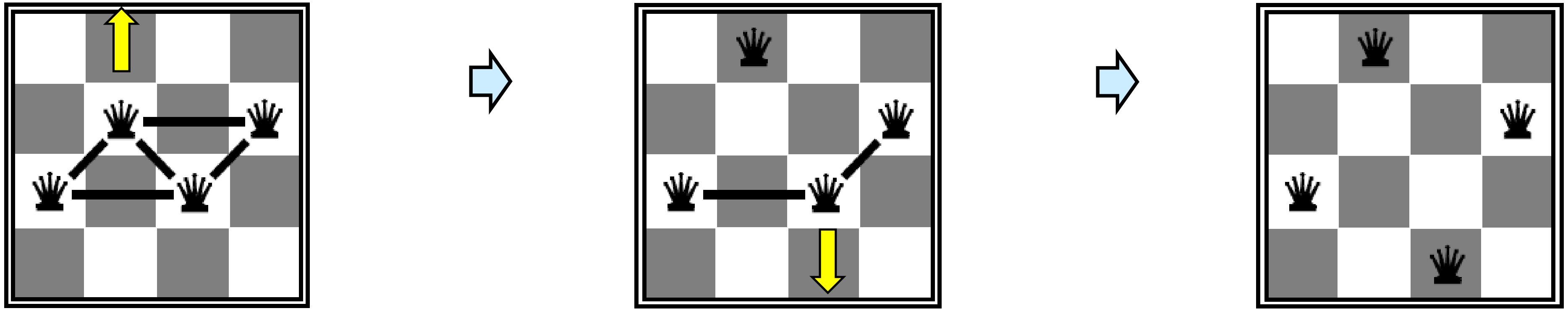
Example: *n*-queens problems *n*皇后问题

- To illustrate hill climbing, we will use the *n*-queens problem. Local search typically use a complete-state formulation.

  为了举例说明爬山法，我们将选用*n*皇后问题。局部搜索通常采用完整状态形式化。

- Put all *n* queens on an *n* x *n* board. Each time move a queen to reduce number of conflicts, to be with no two queens on the same row, column, or diagonal.

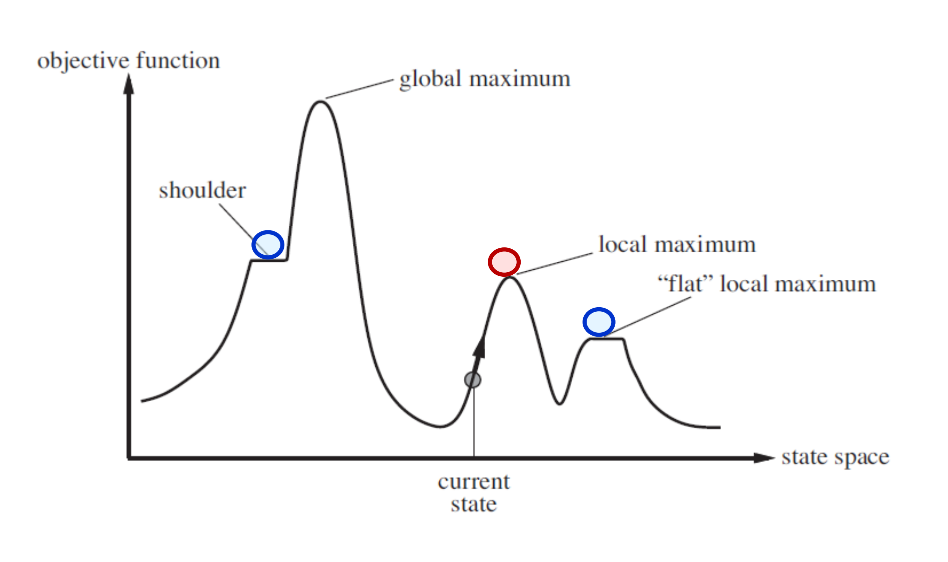
  把n个皇后放在*n* x *n*的棋盘上。每次移动一个皇后来减少冲突数量，使得没有两个皇后在同一行、同一列、或同一对角线上。

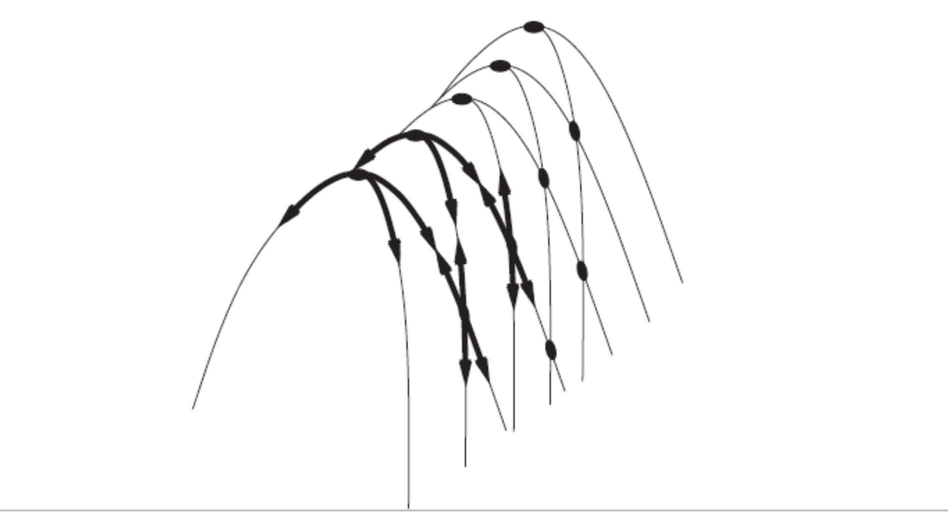


(a) $h = 5$ (b) $h = 2$ (c) $h = 0$

Weaknesses of Hill-Climbing 爬山法的弱点

- It often gets stuck for the three reasons:
  它在如下三种情况下经常被困：
  - **Local maxima** 局部最大值
    higher than its neighbors but lower than global maximum.
    高于相邻节点但低于全局最大值。
  - **Plateaux** 高原
    can be a flat local maximum, or a shoulder.
    可能是一个平坦的局部最大值，或山肩。
  - **Ridges** 山岭
    result in a sequence of local maxima that is very difficult to navigate.
    结果是一系列局部最大值，非常难爬行。





Variants of Hill-Climbing 爬山法的变型

- ☐ Stochastic hill-climbing 随机爬山法
  - It chooses at random among uphill moves; the probability of selection can vary with the steepness of uphill move.
  - This usually converges more slowly than steepest ascent.

  在向上移动的过程中随机选择；选择的概率随向上移动的斜度而变化。与最陡爬坡相比，收敛速度通常较慢。

- ☐ First-choice hill-climbing 首选爬山法
  - It implements stochastic hill climbing by generating successors randomly until one is generated that is better than the current state.
  - This is a good strategy when a state has many of successors.

  它通过随机生成后继点来实现随机爬山法，直到生成一个比当前状态好的点。当某个状态有许多后继时，用此策略为好。

Variants of Hill-Climbing 爬山法的变型

- Random-restart hill-climbing 随机重启爬山法
  - It conducts a series of hill-climbing searches from randomly generated initial states, until a goal is found.
  - It is trivially complete with probability approaching 1, because it will eventually generate a goal state as the initial state.
  - If each hill-climbing search has a probability p of success, then the expected number of restarts required is $1/p$.

它好于其它爬山搜索方法，从随机生成的初始状态直到找到目标。它十分完备，概率逼近1，因为最终它将生成一个目标状态作为初始状态。如果每次爬山搜索成功的概率为p，则重启需要的期望值是1/p。

It adopts the well-known adage: "If at first you don't succeed, try, try again."

Thank you for your attention!

PoAI