Foundation of Artificial Intelligence

人工智能基础

李翔宇

软件学院

Email: lixiangyu@bjtu.edu.cn

Beyond Classical Search 超越经典搜索

- ☐Classical Search
- ☐Local Search Algorithms
  - （1）Hill-climbing search（爬山搜索）
  - （2）Local beam search（局部束搜索）
  - （3）Simulated annealing search（模拟煺火搜索）
  - （4）Genetic algorithms（遗传算法）

Classical Search 经典搜索

- In previous chapter, we addressed a single category of problems, where the solution is a sequence of actions with following features:

  上一章，我们讨论了一个单一类别的问题，其解决方案是具有如下特点的一系列动作：

  - observable, 可观测
  - deterministic, 确定性
  - known environments. 已知环境

- In this chapter, we will discuss the **local search** algorithms, evaluating and modifying one or more current states rather than systematically exploring paths from an initial state.

  本章我们将讨论：局部搜索算法，考虑对一个或多个状态进行评价和修改，而不是系统地探索从初始状态开始的路径。

Classical Search **经典搜索**

- The search algorithms that we have seen so far are designed to explore search spaces systematically， they are called **Classic Search**.

  我们前面介绍过的搜索算法都系统地探索空间，称为经典搜索算法。

- This systematicity is achieved by keeping one or more paths in memory and by recording which alternatives have been explored at each point along the path.

  这种系统性是通过将一条或多条路径保存在内存中，并在路径的每个点上记录已经探索过哪些备选方案来实现的。（record how to chose a node to be expanded）

- When a goal is found, the *path* to that goal also constitutes a ***solution*** to the problem.

  当找到目标时，到达此目标的路径就是这个问题的一个解。

- In many problems, however, the path to the goal is irrelevant to the solution.

  然而在许多问题中，到达目标的路径与解是无关的。

Classical Search 经典搜索

- For example, in the 8-queens problem, what matters is the final configuration of queens, not the order in which they are added.

  例如，在八皇后问题中，重要的是最终皇后在棋盘上的布局，而不是皇后加入的先后次序。

- The same general property holds for many important applications such as integrated-circuit design, factory-floor layout, job-shop scheduling, automatic programming, telecommunications network optimization, vehicle routing, and portfolio management.

  许多重要的应用都具有这样的性质，例如集成电路设计、工厂场地布局、作业车间调度、自动程序设计，电信网络优化、车辆寻径和文件夹管理。

Local Search Algorithms 局部搜索算法

◆In many optimization problems, the path to the goal does not matter; the goal state itself is the solution.

在许多优化问题中，到达目标的路径是无关紧要的；目标状态本身就是解决方案。

◆In such cases, we can use a different class of algorithms that do not worry about paths at all. That is local search algorithm.

在此情况下，我们可以采用一种不同类型的搜索算法，这类算法不关心路径，它就是局部搜索算法。

◆Local search algorithms operate using a single current node (rather than multiple paths), and generally move only to neighbors of that node.

局部搜索算法从单个当前节点（而不是多条路径）出发，通常只移动到它的邻近状态。

◆Typically, the paths followed by the search are not retained.

通常，不保留搜索路径。

Local Search Algorithms

◆Basic idea: in the search process, always search towards the direction closest to the target.

基本思想：在搜索过程中，始终向着离目标最接近的方向搜索。

◆The goal can be the maximum or the minimum.

目标可以是最大值，也可以是最小值

◆Local search algorithms have two key advantages:

局部搜索算法有如下两个主要优点：

- ➢use very little memory; 使用很少的内存；
- ➢can find reasonable solutions in large or infinite (continuous) state spaces. 在大的或无限（连续）状态空间中，能发现合理的解。

Local Search Algorithms

- Methods of Local Search
  - Hill-climbing search（爬山搜索）
  - Local beam search（局部束搜索）
  - Simulated annealing search（模拟煺火搜索）
  - Genetic algorithms（遗传算法）

Hill-climbing Search 爬山搜索

◆ 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。

◆Hill-climbing Search is simply a loop that continually moves in the direction of increasing value—that is, uphill.

爬山搜索是简单的循环过程，不断向值增加的方向持续移动—— 即，登高。

◆It terminates when it reaches a “peak” where no neighbor has a higher value.

算法在到达一个“峰顶”时终止，邻接状态中没有比它值更高的。

◆The algorithm does not maintain a search tree, so the data structure for the current node need only record the state and the value of the objective function.

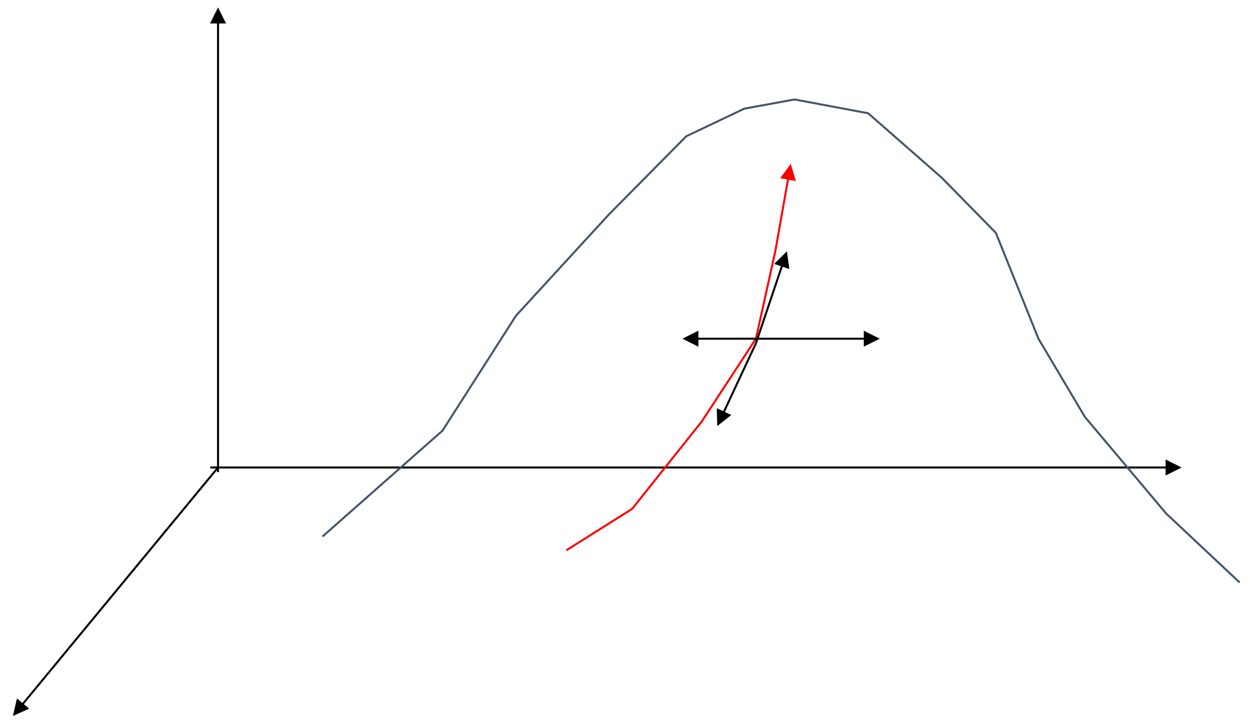
算法不维护搜索树，因此当前结点的数据结构只需要记录当前状态和目标函数值。

◆Hill climbing does not look ahead beyond the immediate neighbors of the current state. 爬山法不会考虑与当前状态不相邻的状态。

（1）Hill-climbing search

- It is sometimes called greedy local search, because it grabs a good neighbor state without thinking ahead about where to go next.
  爬山法有时被称为贪婪局部搜索，因为它总是选择邻居中状态最好的一个，而不考虑下一步该如何走。
- It turns out hat greedy local search algorithms often perform quite well.
  贪婪局部搜索算法往往很有效。
- It often makes rapid progress toward a solution, because it is usually quite easy to improve a bad state.
  爬山法很快朝着解（目标状态）的方向进展，因为它可以很容易地改善一个不良状态。

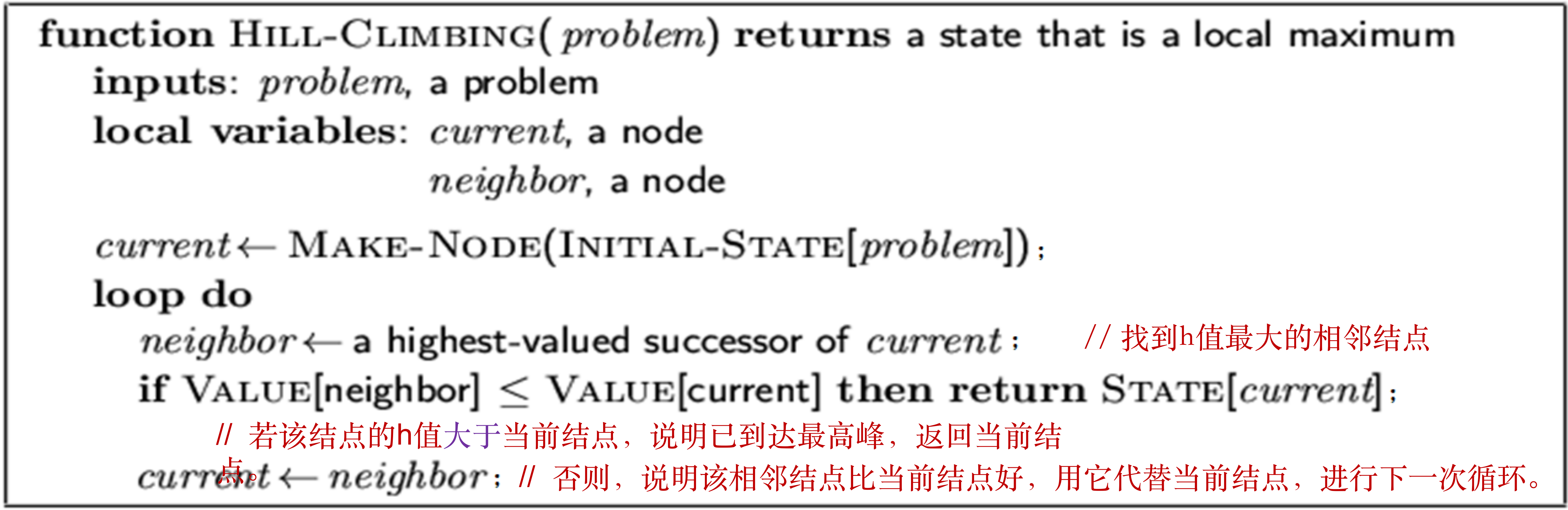
Hill-climbing Search 爬山搜索



If the peak is taken as the goal, h(n) to represent the height difference between the peak and the current position n, the algorithm is equivalent to always climbing towards the top of the mountain. Under the condition of single peak, it will be able to reach the peak.

如果把山顶作为目标，h(n)表示山顶与当前位置n之间的高度差，则该算法相当于总是登向山顶。在单峰的条件下，必能到达山顶。

Hill-climbing search



```
function HILL-CLIMBING(problem) returns a state that is a local maximum
    inputs: problem, a problem
    local variables: current, a node
                     neighbor, a node

    current ← MAKE-NODE(INITIAL-STATE[problem]);
    loop do
        neighbor ← a highest-valued successor of current;     // 找到h值最大的相邻结点
        if VALUE[neighbor] ≤ VALUE[current] then return STATE[current];
            // 若该结点的h值大于当前结点，说明已到达最高峰，返回当前结点。
        current ← neighbor; // 否则，说明该相邻结点比当前结点好，用它代替当前结点，进行下一次循环。
```

At each step the current node is replaced by the best neighbor; in this version, that means the neighbor with the highest VALUE, but if a heuristic cost estimate h is used, we would find the neighbor with the lowest h.
在每一步，当前结点都会被它的最佳相邻结点所代替；这里，最佳相邻结点是指VALUE 最高的相邻结点，但是若使用启发式代价评估函数h， 我们要找的就是h 最低的相邻结点。

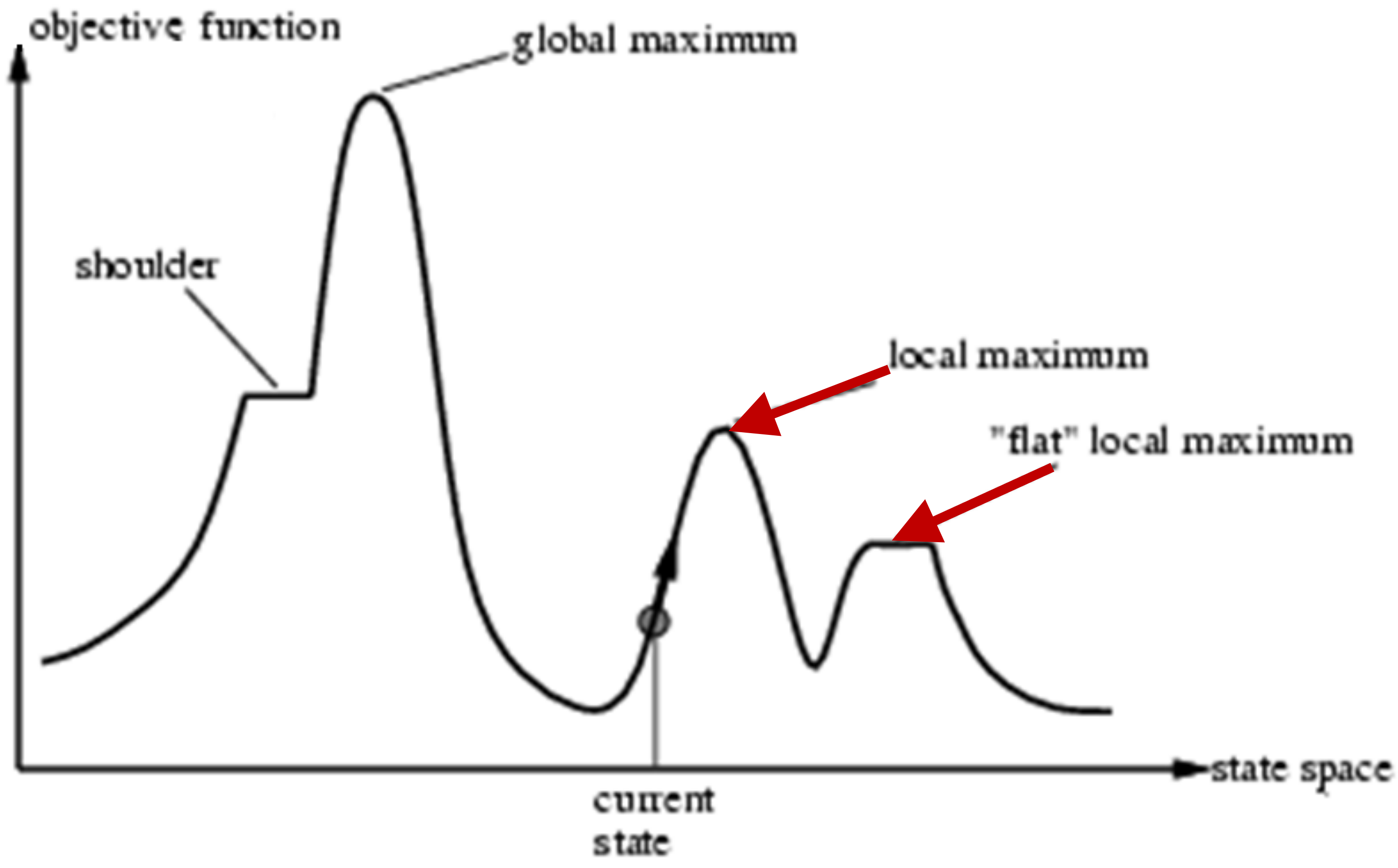
Local Maxima 局部最大值

- A local maximum is a peak that is higher than each of its neighboring states but lower than the global maximum.

  局部极大值是一个比它的每个邻接结点都高的峰顶，但是比全局最大值要小。

- Hill-climbing algorithms that reach the vicinity of a local maximum will be drawn upward toward the peak but will then be stuck with nowhere else to go.

  爬山法算法到达局部极大值附近就会被拉向峰顶，然后就卡在局部极大值处无处可走。



Simulated Annealing 模拟退火

- The innermost loop of the simulated-annealing algorithm is quite similar to hill climbing.
  模拟退火算法的内层循环与爬山法类似。
- Instead of picking the best move, however, it picks a random move.
  只是它没有选择最佳移动，选择的是随机移动。
- If the move improves the situation, it is always accepted.
  如果该移动使情况改善，该移动则被接受。
- Otherwise, the algorithm accepts the move with some probability less than 1.
  否则，算法以某个小于1 的概率接受该移动。
- The probability decreases exponentially with the “badness” of the move—the amount ΔE by which the evaluation is worsened.
  随着移动导致状态“变坏”，概率会成指数级下降——评估值ΔE 变坏。

Simulated Annealing 模拟退火

- The probability also decreases as the “temperature” T goes down: “bad” moves are more likely to be allowed at the start when T is high, and they become more unlikely as T decreases.

  这个概率也随“温度”T的 降低而下降：开始T高的时候可能允许“坏的”移动，当T降低时则不可能允许“坏的”移动。

- If the schedule lowers T slowly enough, the algorithm will find a global optimum with probability approaching 1.

  如果调度让温度T下降得足够慢，算法找到全局最优解的概率接近于1。

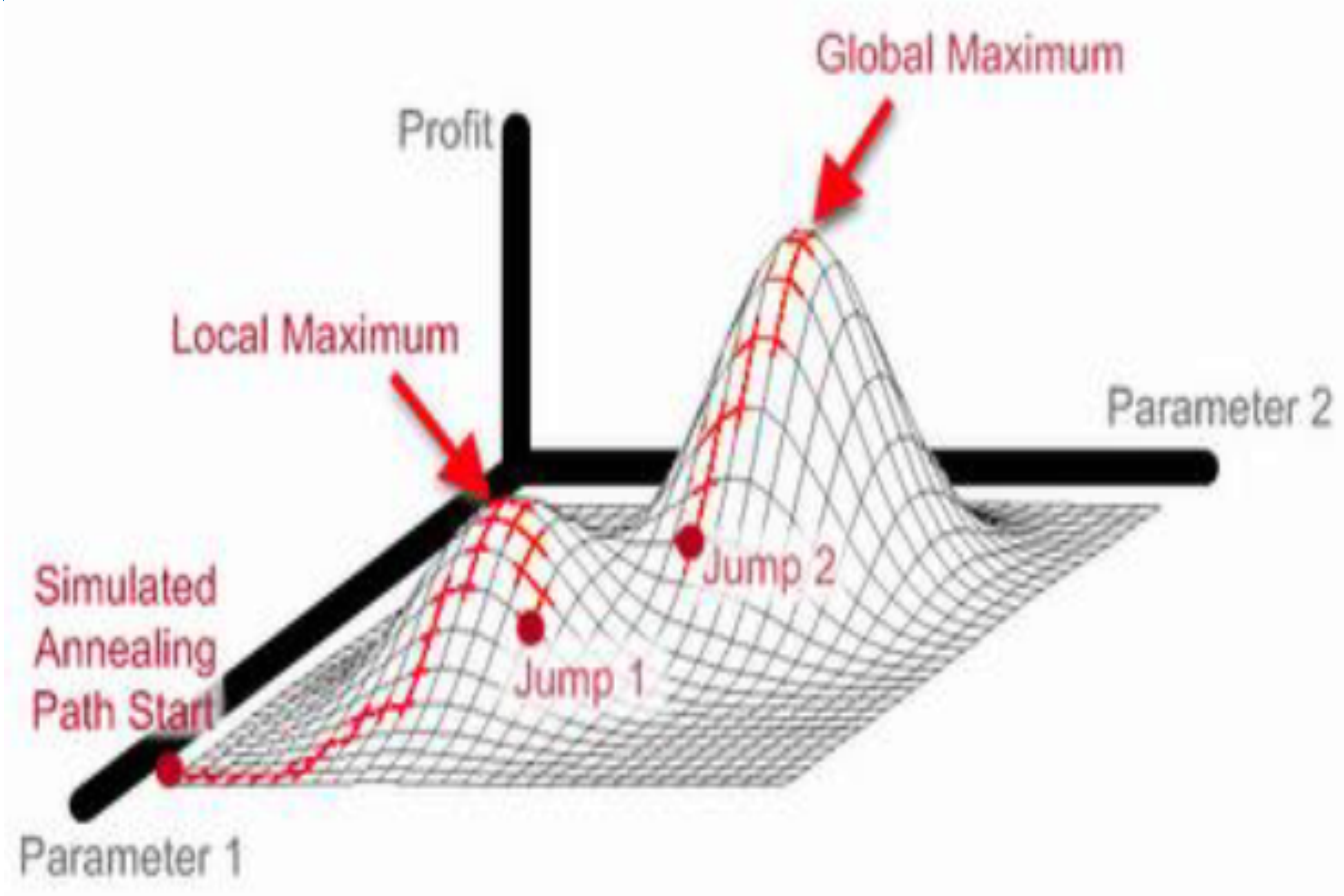
- Simulated annealing was first used extensively to solve VLSI layout problems in the early 1980s. It has been applied widely to factory scheduling and other large-scale optimization tasks.

  模拟退火在20 世纪80 年代早期广泛用于求解VLSI 布局问题。现在它已经广泛地应用于工厂调度和其他大型最优化任务。

Simulated Annealing 模拟退火

- Simulated annealing is a probabilistic method to approach the global optimal solution, which was published in 1953.
  模拟退火是一种逼近全局最优解的概率方法，发表于1953年。

- The simulated annealing algorithm, a version of stochastic hill climbing where some downhill moves are allowed. Downhill moves are accepted readily early in the annealing schedule and then less often as time goes on.
  模拟退火算法，允许下山的随机爬山法。在退火初期下山移动容易被采纳，随时间推移下山的次数越来越少。



Simulated annealing search

```
function SIMULATED-ANNEALING(problem, schedule) returns a solution state
  inputs: problem, a problem
          schedule, a mapping from time to "temperature"
  current ← MAKE-NODE(problem.INITIAL-STATE)
  for t = 1 to ∞ do
    T ← schedule(t)
    if T = 0 then return current
    next ← a randomly selected successor of current
    $\Delta E$ ← next.VALUE – current.VALUE
    if $\Delta E > 0$ then current ← next
    else current ← next only with probability $e^{\Delta E/T}$
```

Local Beam Search 局部束搜索

◆Keeping just one node in memory might seem to be an extreme reaction to the problem of memory limitations.

虽然内存有限，但在内存中只保存一个结点又有些极端。(爬山法只记录当前状态及其目标函数值)

◆The local beam search algorithm keeps track of k states rather than just one.

局部束搜索算法记录k 个状态而不是只记录一个。

◆It starts with k randomly generated states

它从k个随机生成的状态开始。

◆ At each step, all the successors of all k states are generated.

在每一步，生成所有k个状态的所有后继状态。

◆If any one is a goal state, stop; else select the k best successors from the complete list and repeat.

如果其中有一个是目标状态，则算法停止；否则从整个后继列表中选择k个最佳后继，重复这个过程。

Genetic Algorithms 遗传算法

◆The elements of genetic algorithms was introduced in 1960s.Became popular through the work of John Holland in early 1970s, and particularly his book Adaptation in Natural and Artificial Systems(1975).

遗传算法的一些要素是1960年代提出的。通过约翰·霍兰在1970年代早期的工作，尤其是他的《神经元与人工系统的适应性》（1975年）一书，使得遗传算法流行起来。

◆It is a search heuristic that mimics the process of natural selection.

它是一种模仿自然选择过程的搜索启发式算法。

◆In this algorithm, successor states are generated by combining two parent states rather than by modifying a single state. It is dealing with sexual reproduction rather than asexual reproduction.

在该算法中，后继节点是由两个父辈状态的组合而不是修改单一状态生成的。其处理过程是有性繁殖，而不是无性繁殖。

Genetic Algorithms （GA）遗传算法

- Genetic algorithms belong to the larger class of evolutionary algorithms. 遗传算法属于进化算法这个大分类。
- The algorithms generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

  该算法采用自然进化所派生的技法来生成优化问题的解，例如：遗传、突变、选择、以及杂交。
- It begin with a set of k randomly generated states, called the population. 该算法开始时具有一组k个随机生成的状态，称其为种群。

Genetic Algorithms （GA）遗传算法

- Each state, or individual, is represented as a string over a finite alphabet, most commonly, a string of 0s and 1s.

  每个状态，或称为个体，表示为有限字母表上的一个字符串，通常是0和1的字符串。

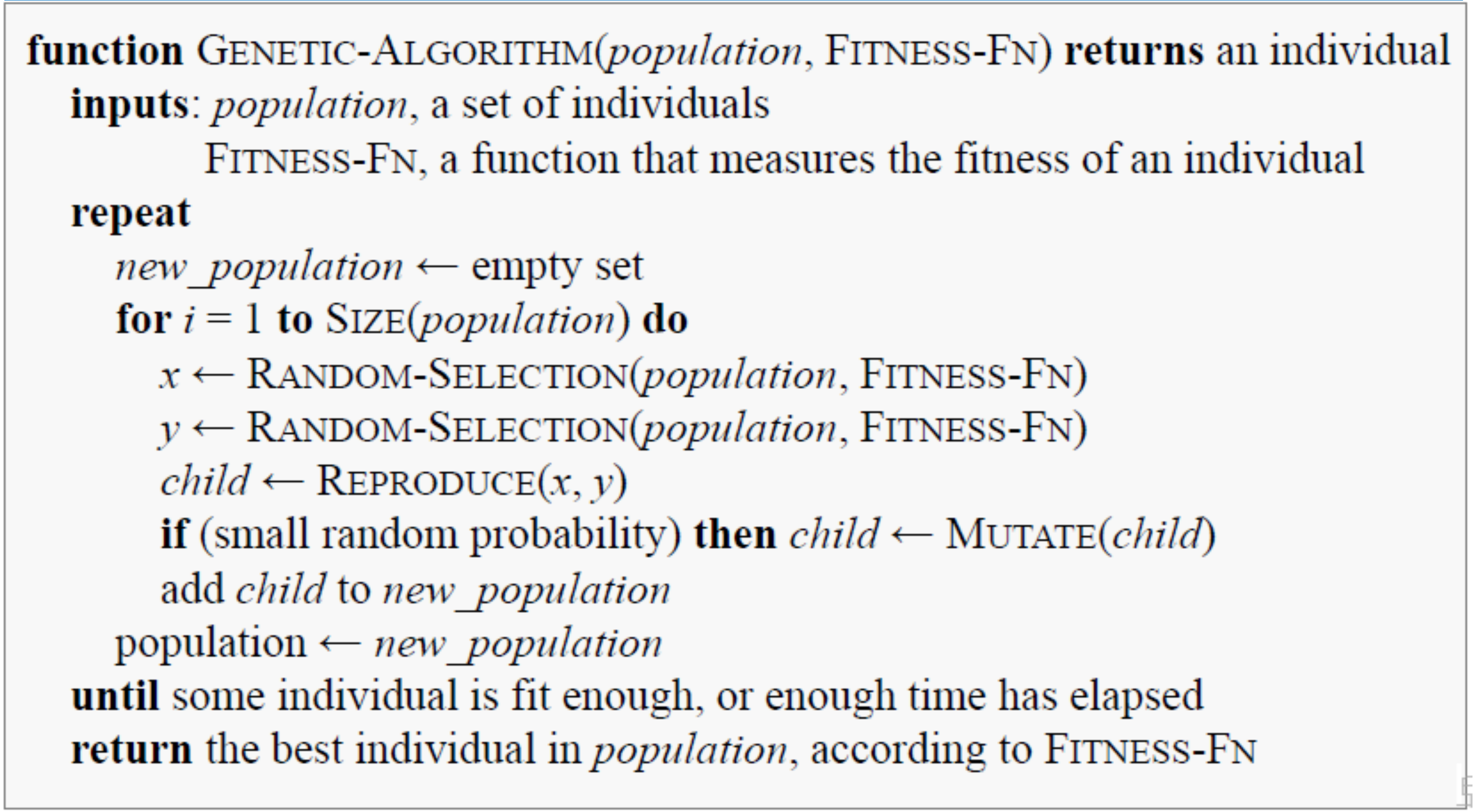
- Each state is rated by the objective function, or (in GA terminology) the fitness function.

  每个状态都由它的目标函数或（用遗传算法术语）适应度函数给出评估值。

- A fitness function should return higher values for better states.

  对于好的状态，适应度函数应返回较高的值。

The Genetic Algorithm



```
function GENETIC-ALGORITHM(population, FITNESS-FN) returns an individual
  inputs: population, a set of individuals
          FITNESS-FN, a function that measures the fitness of an individual
  repeat
    new_population ← empty set
    for $i$ = 1 to SIZE(population) do
      $x$ ← RANDOM-SELECTION(population, FITNESS-FN)
      $y$ ← RANDOM-SELECTION(population, FITNESS-FN)
      child ← REPRODUCE($x$, $y$)
      if (small random probability) then child ← MUTATE(child)
      add child to new_population
    population ← new_population
  until some individual is fit enough, or enough time has elapsed
  return the best individual in population, according to FITNESS-FN
```

Applications of Genetic Algorithms 遗传算法的应用

bioinformatics	🟥	生物信息学
computational science	🟥	计算科学
engineering	🟥	工程
economics	🟥	经济学
chemistry	🟥	化学
manufacturing	🟥	制造
mathematics	🟥	数学
physics	🟥	物理
phylogenetics	🟥	种系遗传学
pharmacometrics	🟥	定量药理学

Thank you !