**4.1局部搜索算法和最优化问题**

不关心路径，专注于寻找目标 或根据目标函数找到最佳状态

它不记录路径，也不记录已达状态集。

它们**不是系统性的**——可能永远不会探索问题的解实际所在的那部分搜索空 间

两个主要优点：（1）使用很少的内存；（2）通常可以在系统性算法不适用的大型或无限状态空间中找到合理的解。

**4.1.1爬山法**

最陡上升/贪婪局部搜索：只选择**邻居**中状态最好的一个，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

遇到平台区时，允许横向移动。可以限制连续横向移动的次数，如在100次连续横向移动之后停止。

**爬山法变体：**

1. 随机爬山法（stochastic hill climbing）

在**上坡行动中**随机选择一个；被选中的**概率**随着上坡陡度的变化而变化。

**收敛得更慢**，但**可能找到更好的解**。

1. 首选爬山法（first-choice hill climbing）

通过不断**随机地生成后继**直到生成一个**比当前状态更好**的后继为止来实现随机爬山。

当一个状态**存在众多（如数千个）后继**时，这是一个很好的策略。

1. 随机重启爬山法（random-restart hill climbing）

它从**随机生成的初始状态**开始，执行一系列爬山搜索，直到找到目标。

算法**完备**的**概率为1**

**4.1.2模拟退火搜索**

随机移动。如果该移动使情况改善，该移动则被接受。否则，算法以某个小于1的概率（, ΔE评估值变坏的量）接受该移动。

如果移动导致状态“变坏”，概率则成指数级下降。这个概率也随“温度”T降低而下降：开始T高的时候可能允许坏的移动，T越低则越不可能发生。

如果调度**让T下降得足够慢，算法找到全局最优解的概率逼近于 1。**

**4.1.3局部束搜索**

它从k个随机生成的状态开始。每一步全部**k**个状态的所有后继状态全部被生成。如果其中有一个是目标状态，则算法停止。否则，它从整个后继列表中选择**k个最佳**的后继，重复这个过程。

**有用的信息在并行的搜索线程之间传递**

**4.1.4遗传算法**

（1）遗传算法也是从**k**个随机生成的状态开始，我们称之为种群。每个状态，或称个体，用一个有限长度的字符串表示

（2）产生下一代状态：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

（3）按照适应度给出的概率随机地选择两对进行繁殖。

（4）父串在杂交点上进行杂交而创造出后代

（5）后代串每个位置都会按照某个小的独立概率随机变异

**4.2连续空间中的局部搜索**

**4.3使用不确定动作的搜索**