

# Part III-B: Artificial Intelligence Outline

Lecture by 熊庆宇

Note by THF

2024 年 10 月 21 日

## 目录

0.1 启发式搜索 . . . . .	2
---------------------	---

## Lecture 8

10.17

**Notation.** DFS: Deep-First Search

优先朝长节点扩展：先进入开放表的节点先扫描

优点：搜索空间可以远小于宽度优先

缺点：忽略深度

修正：加入深度界限，在已知目标节点的深度范围时限制搜索深度

最坏情况： $o(n)$

应用：状态表 = 树状图

**Notation.** 等代价搜索/Dijkstra 算法：

BFS 的一种推广

$g(n)$  代表从初始节点到节点  $n$  的代价

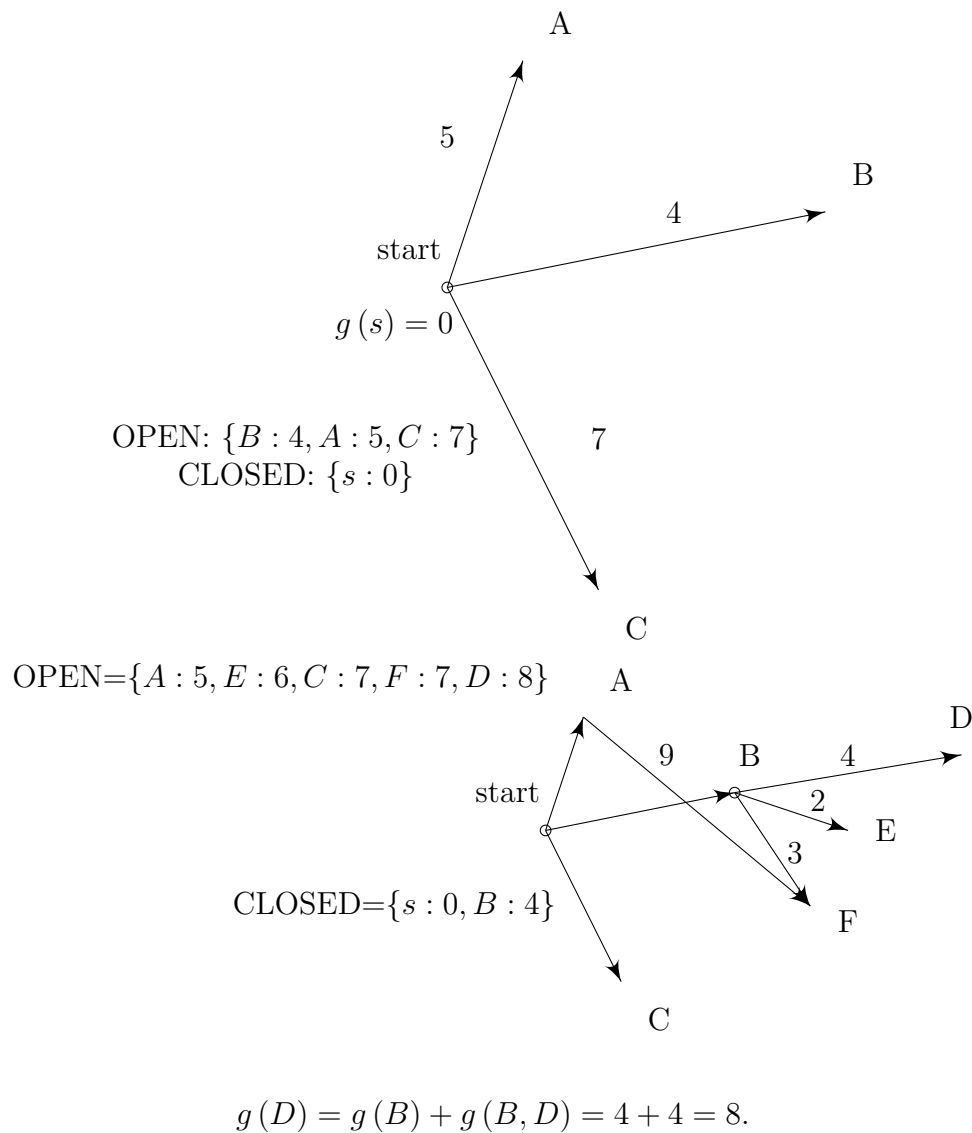
$c(n_1, n_2)$  表示从  $n_1$  到  $n_2$  的代价

$g(n_2) = g(n_1) + c(n_1, n_2)$

优点：加入了状态图中的路径长短元素（走一步看一步），以等代价选择下一节点的选择

## Lecture 9

10.21



## 0.1 启发式搜索

**Notation.** 盲目搜索的不足：效率低、组合爆炸、产生大量无用节点

**Notation.** 启发式信息：与具体问题求解过程有关的，指导搜索过程朝最可能前进方向的数据

**Notation.** A 算法：

引入估价函数:  $f(n) = g(n) + h(n)$

$g(n)$ : 从起始状态到当前状态已实际付出的代价

$h(n)$  从当前状态到目标状态的估计代价 (启发函数)

**Example.** 错位个数: 与目标状态的比较差别

2	8	3	$\xrightarrow[\text{错位个数: 4 (不包含空格)}]{\text{目标状态}}$	1	2	3
1	6	4		8		4
7		5		7	6	5

$$g(n) = 0 \quad h(n) = 4.$$

可得  $f(n) = g(n) + h(n) = 4$

类似于等代价算法, 通过比较估价函数值即可减少遍历节点数

**Notation.**  $A^*$  算法: 对函数进行限定, 使其一定可以找到最优解

$$A^* = g(n) + h(n).$$

$g(n)$  为起点到  $n$  点已走过距离

$g^*(n)$  是起点到  $n$  点的最短路径

$g(n)$  是对  $g^*(n)$  的估计

$h(n)$  为引导从  $n$  点到目的地的参照距离, 一般为欧氏距离  $L_2(\mathbf{x}_i, \mathbf{x}_j)$

$h^*(n)$  为从  $n$  点到目的地的实际最短距离,  $h(n) \leq h^*(n)$

**Example.** 百度地图: 一直有一条红线引导方向, 该红线即是  $h(n)$

确定的路线为绿色, 为  $h^*(n)$

**Example.** 八数码难题:  $h_1(n)$  表示不在位置上的数字数量

$h_2(n)$  表示节点  $n$  到目标位置的曼哈顿距离之和

易得  $0 \leq h(n) \leq h_1(n) \leq h_2(n)$