

文章目录

1. A* 树搜索
2. A* 图搜索
3. 一致性与可采纳性

对一个图（有向图），要找到从起点到终点的一条路径，既可以用图搜索算法，也可以用树搜索算法。

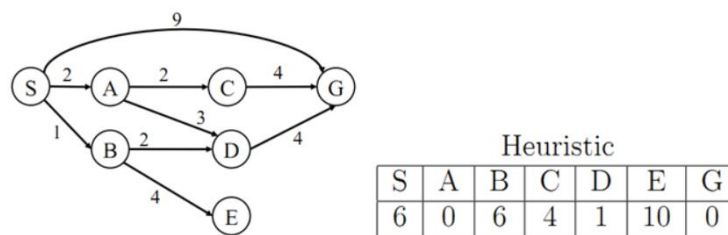
图搜索算法 **不允许** 重复访问结点。

树搜索算法允许重复访问结点。

以 A* 搜索为例，启发函数为 $f(n) = g(n) + h(n)$ ， $g(n)$ 是从起点到当前结点走过的路径的代价， $h(n)$ 是当前节点到目标的估计代价。每次 A* 搜索都会搜拓展节点中 f 最小的点。

例子

图一表示一个搜索问题，其中结点表示状态，边上的数字表示状态之间的实际代价，S 为初始结点，G 为终止结点，表格中的数字表示结点到终止结点的估计代价。根据图一回答如下问题：



图一：搜索状态图

- c) 给出 A* 的树搜索路径，要求给出每一步的扩展结点和代价值（5 分）；
- d) 给出 A* 的图搜索路径，要求给出每一步的扩展结点和代价值（5 分）；

1. A* 树搜索

Algorithm progression:

Path expanded	Fringe (ordered by path + heuristic cost)
S	S-A(2+0) S-B(1+6) S-G(9+0)
S-A	S-A-D(5+1) S-B(1+6) S-A-C(4+4) S-G(9+0)
S-A-D	S-B(1+6) S-A-C(4+4) S-G(9+0) S-A-D-G(9+0)
S-B	S-B-D(3+1) S-A-C(4+4) S-G(9+0) S-A-D-G(9+0) S-B-E(5+10)
S-B-D	S-B-D-G(7+0) S-A-C(4+4) S-G(9+0) S-A-D-G(9+0) S-B-E(5+10)
S-B-D-G	S-A-C(4+4) S-G(9+0) S-A-D-G(9+0) S-B-E(5+10)

Fringe 中包含了当前拓展出的所有可以走但尚未访问的路径。

需要注意的是第四行选择的 *S-B* 这一条路径，它可以拓展出 *S-B-D* 这一条路径放入 *Fringe*，树搜索允许重复访问结点。

2. A* 图搜索

Algorithm progression:

Path expanded	Closed list	Fringe (ordered by path + heuristic cost)
S	S	S-A(2+0) S-B(1+6) S-G(9+0)
S-A	S A	S-A-D(5+1) S-B(1+6) S-A-C(4+4) S-G(9+0)
S-A-D	S A D	S-B(1+6) S-A-C(4+4) S-G(9+0) S-A-D-G(9+0)
S-B	S A D B	S-A-C(4+4) S-G(9+0) S-A-D-G(9+0) S-B-E(5+10)
S-A-C	S A D B C	S-A-C-G(8+0) S-G(9+0) S-A-D-G(9+0) S-B-E(5+10)
S-A-C-G	S A D B C G	S-G(9+0) S-A-D-G(9+0) S-B-E(5+10)

要注意的是第四行选择的 *S-B* 这一条路径，*D* 已经在之前访问过了并加入到了 *Closed List* 中，所以 *S-B* 只能拓展出 *S-B-E* 这一条路径。

3. 一致性与可采纳性

针对启发函数 f 而言，

如果 $f(n)$ 可采纳，那么要满足：

对于每个结点 n ， $f(n) < h^*(n)$ 。其中 $h^*(n)$ 是到达目标节点的真实代价。

这意味着可采纳的启发函数绝对不会高估到达目标节点的代价，因此它是最优的。

如果启发函数是可采纳的，那么A*使用树搜索是最优的。

📖 文章知识点与官方知识档案匹配，可进一步学习相关知识

算法技能树 > 首页 > 概览 61681 人正在系统学习中