什么是 A*算法

[马少平，王东]

在现实生活中，我们经常会遇到求解最佳路径问题，比如导航软件要求从 S 点到 G 点的最短路径，A*算法就是一个有效的求解最佳路径的搜索算法。

假设我们在地图上求从 S 出发到达目标 G 的最短路径，每一个路口可以视为一个状态，最短路径问题就是在地图上找出一个从 S 到达 G 的状态序列，使得沿着这一状态序列行走的距离最短。该状态序列称为从 S 到 G 的最短路径，寻找最短路径的过程称为状态搜索。

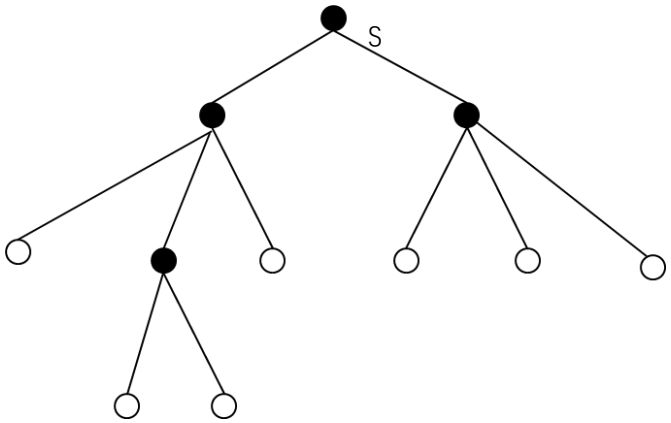

图 1. 一个局部状态搜索图

图 1 给出一个状态搜索图的样例，其中每个节点代表一个状态，即路径中的一个路口，两个状态间的连线表示直通道路。为了找到一条从 S 到达 G 的路径，我们每次从上图中选择一个叶节点进行扩展，直到找到目标节点 G 为止。

问题是，在搜索图中有很多叶节点，究竟应该对哪个节点进行扩展呢？一个直观的方案是，如果某个叶节点 n 距离初始节点 S 的距离再加上节点 n 到目标节点 G 的最小距离之和最小，那么该节点处在最短路径上，应该优先扩展。我们用 g(n)表示 S 到 n 的距离，用 h(n)表示 n 到 G 的最短路径距离，则从 S 经过 n 到达 G 的总距离 f(n)为：

$$f(n) = g(n) + h(n)$$

如果我们选择 f(n)最小的那的叶节点进行扩展，将保证搜索效率最高。

这里 g(n)比较简单，可以从当前的状态搜索图中直接计算出来。困难的是 h(n)，因为在搜索过程中，我们还没有找到一条从 n 到 G 的最短路径，因此也就不知道 h(n)是多少。一种可能的方案是估计出一个 h'(n)，并用它来代替实际最短路径 h(n)进行搜索。问题是，这样是否可以得到一条合理的、甚至是最短的路径呢？答案是肯定的。

我们首先考虑某种路径估计方法 h'(n), 基于 h'(n)可以估计经过 n 的最短路径 f'(n)=g(n)+h'(n)。算法从初始节点 S 开始，每次选择一个 f'(n)值最小的叶节点进行扩展，直到扩展出目标 G 且 f'(G) 在所有叶节点中取值最小为止。在搜索过程中，如果遇到多条路径到达同一个节点的

情况，需要更新从 S 到达该节点的最短路径及最短路径估计值 f'(n)。

上述算法被称为 A 算法。在 A 算法中，对 h'(n)没有明确限制，只要符合直觉即可，因此 h'(n)可能比 h(n)小，也可能比 h(n)大，其中 h(n)为 n 到 G 的最短路径。如果对 h'(n)加以限制，使得对于任何一个叶节点 n，总有 h'(n)<=h(n)，那么找到的路径一定是最短的，此时 A 算法被称为 A*算法。

如何证明这一点呢？让我们考虑搜索结束时的所有 f'(n)，其中 n 为任意一个叶节点。因为 f'(G)最小，必有 f'(G)<=f'(n)；再考虑条件 h'(n) <= h(n)，则必有:

$$f'(G) <= f'(n) = g(n) + h'(n) <= g(n) + h(n) = f(n)$$

因此 A*算法得到的路径是最短路径，f'(G)是最短路径的距离。

在前述地图搜索任务中，我们可以通过城市的坐标值来计算两个城市之间的直线距离。显然，两点之间的直线距离肯定小于他们之间的实际最短路径距离。因此，利用这一直线距离来计算 h'(n)可以满足 A*算法的条件，所以一定可以搜索到一条最短路径。图 2 给出一个 A*算法执行路径搜索的运行过程。

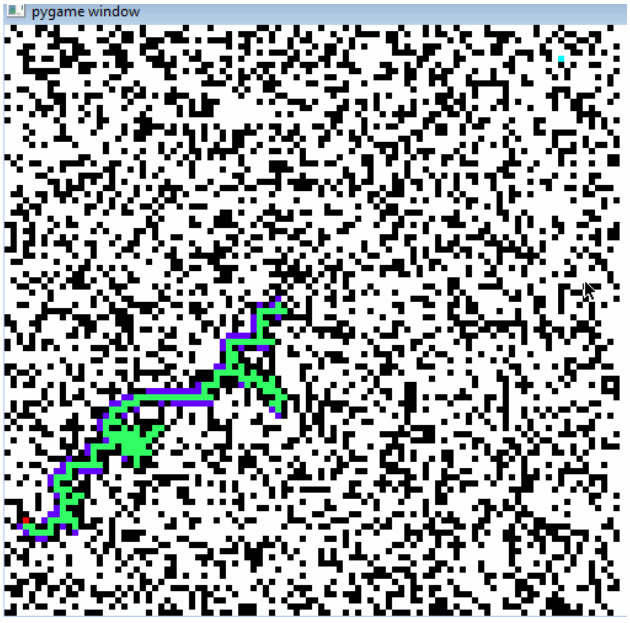

图 2：A*算法搜索过程示意图（动图）[1]

A*算法不限于搜索地图上的最短路径，很多问题，只要可以转化为状态搜索问题，都可以使用 A*搜索，比如：八数码问题、旅行商问题、传教士与野人问题等。

1．An A* path finding algorithm animation, https://imgur.com/gallery/vC71luv