

# A\* 算法估价函数的特性分析

钟 敏

(武汉工程职业技术学院 武汉:430080)

**摘 要** 确定估价函数是 A\* 算法中最关键的问题。在对估价函数特性进行分析的基础上,讨论了它的几个一般构造原则,并简要介绍一些试验经验及若干实例。

**关键词** A\* 算法 估价函数 启发函数 最短路径

**中图分类号**:TP183 **文献标识码**:A **文章编号**:1671-3524(2006)02-0031-03

## 1 引言

A\* 算法是人工智能中的一种启发式搜索算法,它在角色扮演(role play game,以下简称 RPG)游戏中有它很典型的用法,可以说是人工智能技术在游戏应用中的代表。RPG 游戏中都要用鼠标控制人物运动,而且让人物从当前位置走到目标位置应该走最短路径。目前普遍采用 A\* 算法,在不考虑时间和空间限制的情况下,它可以确保找到一条到达目标结点的最佳路径。

A\* 其实是在宽度优先搜索的基础上引入了一个估价函数,每次并不是把所有可展的结点展开,而是利用这个估价函数对所有没有展开的结点进行估价,从而找出最应该被展开的结点,将其展开,直到找到目标结点为止。A\* 算法实现起来并不难,难就难在建立一个合适的估价函数,估价函数构造得越准确,则 A\* 搜索的时间越短。然而建立估价函数还没有严格的方法可循,为此,本文针对这一问题进行了讨论。

## 2 估价函数的特性分析

在 A\* 算法中,结点  $n$  的估价函数  $f(n)$  由以下公式给出:

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

$g(n)$  是起始结点到  $n$  的实际路径代价, $h(n)$  就是  $n$  到目标的最短路径的启发函数。

凡是一定能找到最佳求解路径的搜索算法称为可采纳的(admissible),数学上已严格证明 A\* 算法是可采纳的。其充要条件是:对任意结点  $n$ ,都有  $h(n) \leq h'(n)$ ,  $h'(n)$  是  $n$  到目标的实际最短距离。这时,也称  $h(n)$  是可采纳的。只有应用可采纳的  $h(n)$  的最好优先算法才是 A\* 算法,否则只能算做 A 算法。

如果  $h(n)$  满足  $h(n_1) - h(n_2) \leq c(n_1, n_2)$ ,  $c(n_1, n_2)$  是从任意结点  $n_1$  转移到另一任意结点  $n_2$  的代价,则称  $h(n)$  是相容的。也就是说状态转移时,下界  $h$  的减少值最多等于状态转移的实际代价,其实就是要求  $h$  不能减少得太快。

那么,对于  $n_1$  和它的后继结点  $n_2$ ,

有  $f(n_1) = g(n_1) + h(n_1)$  及  $f(n_2) = g(n_2) + h(n_2)$ ,

又由  $g(n_2) = g(n_1) + c(n_1, n_2)$  得  $f(n_2) = h(n_2) + g(n_1) + c(n_1, n_2)$ 。

如果  $h(n)$  是相容的,则有  $h(n_1) \leq h(n_2) + c(n_1, n_2)$ ,两边同时加上  $g(n_1)$ ,

得  $h(n_1) + g(n_1) \leq h(n_2) + g(n_1) + c(n_1, n_2)$ ,也就是  $f(n_1) \leq f(n_2)$ ,

由此可见,如果启发函数  $h(n)$  相容,则估价函数  $f(n)$  单调递增。也就是说 A\* 算法扩展的结点序列的  $f$  值会是非递减的,那么它最先生成的路径一定是最短的。这样的话,搜索过程中就不再需要 closed 表,只需维护一个已访问结点表,表中的结点都无需再访问。

### 3 估价函数的几个构造原则

估价函数的构造没有什么规律可循,原则上只要有利于问题的求解,可以随意定义。但通过上一节对估价函数特性的分析,我们可以得出结论:启发函数  $h(n)$  必须可采纳,最好还相容。但事实上,并不是所有可采纳的  $h(n)$  都是相容的。

如前所述,估价函数中  $h(n)$  必须可采纳,否则  $A^*$  算法就只能称作  $A$  算法。 $A$  算法不能保证最后的结果是最优的,但它的速度可能是非常快。然而有些 RPG 游戏的实时性要求高,也不一定非要得到最优解。在这种情况下,设计估价函数可不拘泥于  $h(n)$  的可采纳性限制。当然,这种情况比较少。

$h(n)$  可采纳就一定能找到最短路径,但  $h(n)$  与实际值  $h'(n)$  不能差得太远。差得越远,  $A^*$  最后的搜索拓扑就越接近一个完全的宽度优先搜索。最极端的情况:  $h(n) \equiv 0$  时,  $A^*$  就完全退化为宽度优先搜索。那么,  $h(n)$  要尽可能接近  $h'(n)$ 。理论上来说,  $h(n) = h'(n)$  是最好的,估计值就是实际值,但这在实际中是不可能达到的。

再说启发函数  $h(n)$  的信息量问题,所谓  $h(n)$  的信息量,就是在估计一个结点的值时的约束条件,如果信息越多(或说约束条件越多),则估价函数越准,排除的结点越多,那么  $A^*$  性能越好。宽度优先搜索之所以不可取,就是因为它的启发函数  $h(n)$  一点启发信息都没有。但不能不注意到,  $h(n)$  的信息越多,计算量就越大,耗费的时间也就越多。而游戏里通常要设置超时控制的代码,当内存消耗过大或用时过久就退出搜索。那么就应该适当的减小  $h(n)$  的信息量,即减少约束条件,但这样又会导致算法准确性下降,所以设计  $h(n)$  时需要根据具体应用环境来进行综合平衡设计。

### 4 估价函数实例分析

在实际情况中,我们通常是使用  $h(n)$  的实值函数,再通过试验优化。根据这些积累下来的经验合理地构造估价函数,可以得到更好的结果。估价函数的计算方法可以是坐标差的绝对值乘加权值求和,平方或开方以及一些更复杂的。计算两点间的直线距离是最基本的,但缺点是与现实情况相差太

远。另一种简单实用的方法是计算在街区环境中移动的最短路径:

$$h = \text{abs}(x - \text{endx}) + \text{abs}(y - \text{endy});$$

相加点  $(x, y)$  和目标  $(\text{endx}, \text{endy})$  间的垂直距离和水平距离,不考虑障碍物,比采用两点间的直线距离更适合在城区地图上使用。

估价函数与搜索规模也有关系,我们通过求和后再乘以一个系数  $W$  来适应:

$$h = \text{abs}(x - \text{endx}) + \text{abs}(y - \text{endy}); h = h * W$$

$W$  要通过试验来确定,经验表明可以是 8 或者 10。在地形复杂的时候还可以使用坐标差加权后再求和:

$$dx = \text{abs}(x - \text{endx}), dy = \text{abs}(y - \text{endy});$$

$$\text{if } (dx > dy) h = 10 * dx + 6 * dy; \text{ else } h = 10 * dy + 6 * dx;$$

所用的乘 10 和乘 6 是试验证明比较好用的数字。

更值一提的是,以上这些启发函数都具有相容特性,可以简化搜索所使用的数据结构,但它们相对来说都比较简单,仅仅考虑了距离而忽略了方向,在棋盘状的游戏地图中使用价值比较高,但不太适用于地形复杂的导游地图。下面介绍一个启发函数包含了角度和距离两个因素,比较适用于导游地图:

$$h(n_j) = \rho(n_i, n_j) + W \{ D(s, t) d(n_i, n_j) \}$$

其中,  $n_i$  是当前结点,  $n_j$  是当前结点的后继子结点之一。以起始结点和目标结点之间的连线为零度基准线,  $\rho(n_i, n_j)$  表示  $n_i$  与  $n_j$  之间的连线与基准线之间的角度,用弧度来表示。  $d(n_i, n_j)$  为  $n_i$  与  $n_j$  的距离,  $D(s, t)$  为起始结点与目标结点之间的距离,  $W$  为加权系数。

该启发函数引入方位和距离两个因素,加大了启发的信息量,也加快了搜索的过程和准确度。由于篇幅限制,这里只引入一个小规模问题作简单示意,图 1 是一个公园的导游地图示例:

用  $A^*$  算法搜索从 A 点到 K 点的最短路径,使用该估价函数得到的搜索树如图 2(a),使用估价函数  $h = \text{abs}(x - \text{endx}) + \text{abs}(y - \text{endy})$  得到的搜索树如图 2(b):

显然, (b) 扩展了 8 个节点,而 (a) 仅扩展了 6 个。在大规模问题中,前者的优势还要明显些,而且它在时间和空间上都是可以接受的。

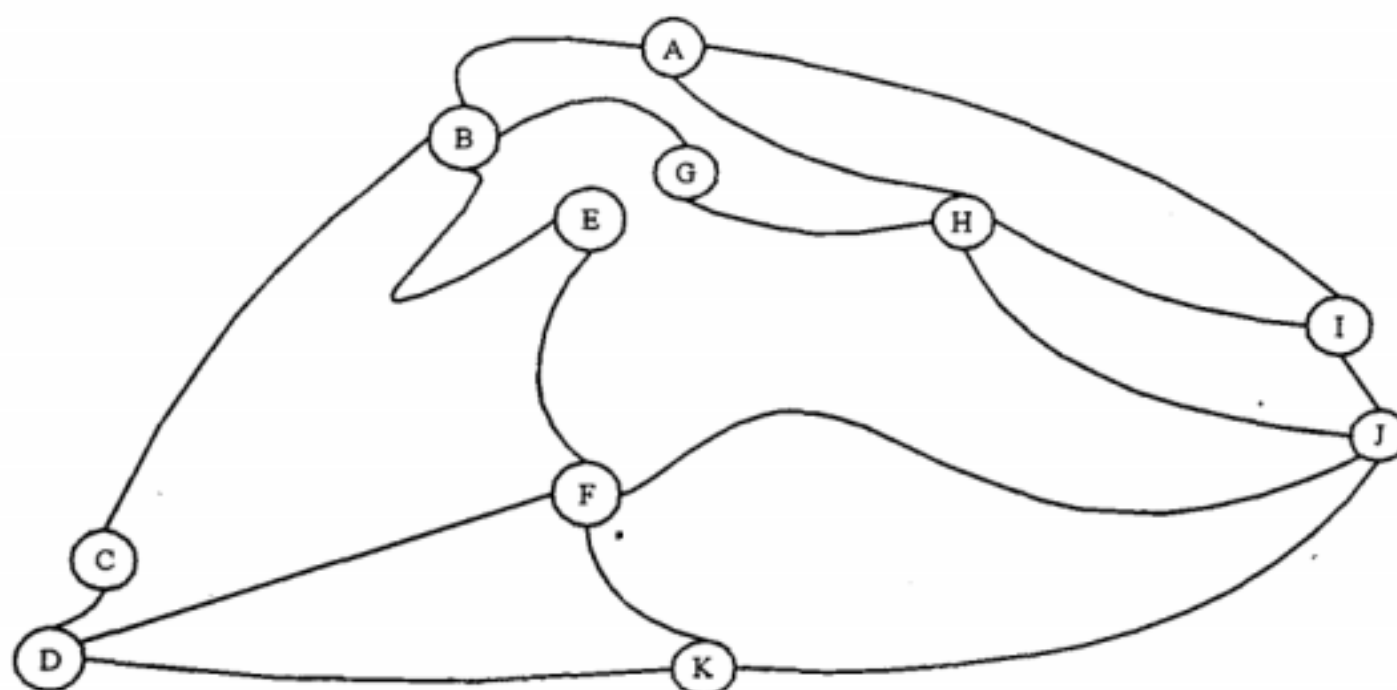


图1 公园导游地图示例

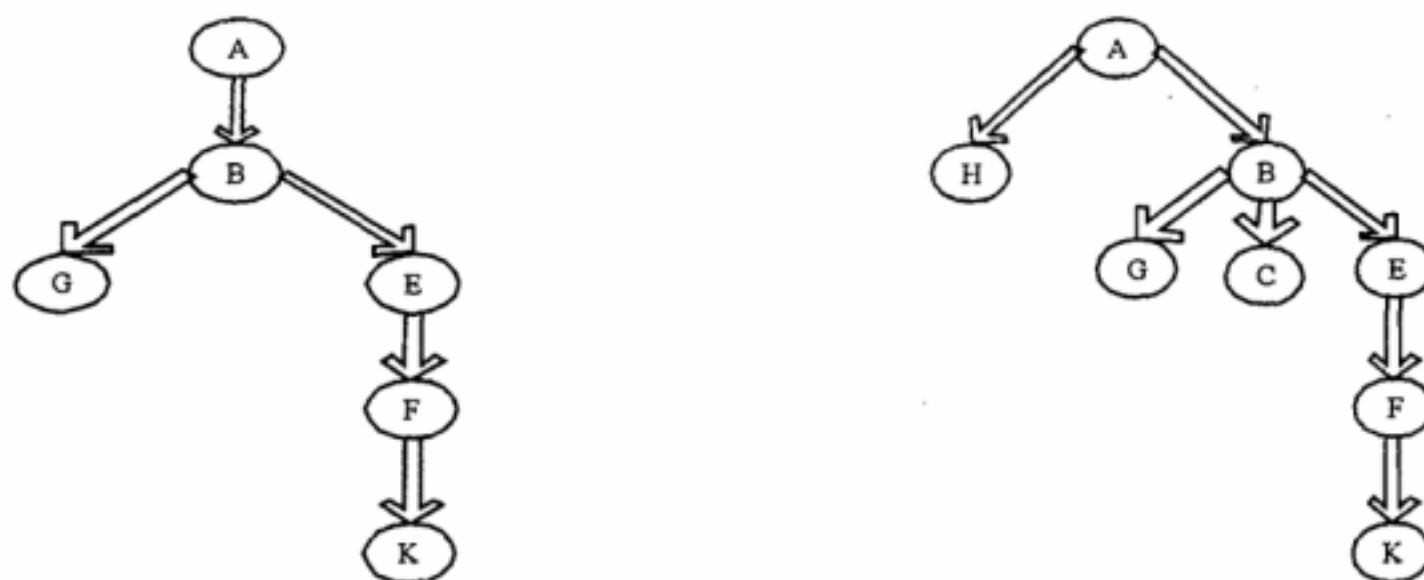


图2 使用A\*算法得到的搜索树

## 5 结束语

本文就估价函数的设计问题进行了初步的探讨,目的是使 A\* 算法能尽快的找到最短路径。然而,用户需要找出的有时候可能不是最短路径,而是满足某些附带要求的尽可能短的路径。A\* 算法这时就需要做一些改进,如引入感兴趣的集。这是有待我们进一步研究的问题。

## 参考文献

- [1] 王德春等. 基于 A\* 算法的舰船最佳航线选择[J]. 青岛大学学报, 2005, 18(4)
- [2] 杨素琼等. 基于 A\* 算法的地图路径搜索的实现[J]. 铁路计算机应用, 2001, 9(4)
- [3] Amit's A\* Pages <http://theory.stanford.edu/~amitp/GameProgramming/>
- [4] 蔡自兴等. 人工智能及其应用[M]第二版. 北京:清华大学出版社, 1996
- [5] 吴泉源等. 人工智能与专家系统[M]. 长沙:国防科技大学出版社, 1995

## Discussion of Designing Cost Function of A\* Algorithm

Zhong Min

**Abstract:** It is a key issue how to design cost function of A\* algorithm. In this paper, the characteristics of cost function are analyzed, several principles on its design discussed. In addition, some experimental results and examples of cost function are introduced.

**Key words:** A\* algorithm, cost function, heuristic function, shortest path

(责任编辑: 栗 晓)