文章编号:1009-038X(2001)05-0531-03

A* 算法中引入前视机制的探讨

朱嘉钢
(江南大学 信息工程学院,江苏 无锡 214036)

摘 要:提出了提高A*算法搜索效率的方法,通过在A*算法中引入前视机制,提高了A*算法的估价函数的质量,减少了平均搜索树宽,提高了搜索效率.理论和仿真都表明这一方法的有效性.

关键词:估价函数;A*算法;人工智能
中图分类号:TP 18 **文献标识码**:A

Mechanism of Introducing Look-ahead Function into A* Algorithm

ZHU Jia-gang
(School of Information Engineering, Southern Yangtze University, Wuxi 214036, China)

Abstract: In this paper, a method for improving the searching efficiency of A* algorithm was proposed. Through introducing look-ahead function into A* algorithm, the evaluating function can be improved and therefore A* algorithm may become more efficient. The effectivity of this method was proved through emulation.

Key words: evaluating function; A* algorithm; artificial intelligence

A*算法是一种利用启发式信息进行搜索的经典的人工智能算法.由于利用了启发式信息,A*算法将搜索引向最有希望成为最佳路径上结点的方向,其搜索效率大大高于盲目搜索算法的搜索效率,具有很大的研究价值和应用价值[3,4].然而,启发式信息来自估价函数,而估价函数的设计往往凭设计者的直觉.如何通过提高h函数的质量来提高A*算法的搜索效率,目前尚未见有关的研究报道.作者探讨通过在A*算法中引入前视机制来提高估价函数的质量,从而提高A*算法搜索效率的方法.

1 A*算法与h函数

A*算法建立在寻找图中最佳路径的算法Graphsearch基础上.如果用一个估价函数 f 在Graphsearch算法的第8步对OPEN表的结点进行排序,f 值小的优先(即OPEN表上具有最小 f 值的的那个结点可作为该趟中的 N,下步从这个结点扩展),其中 $f(n)=g^*(n)+h(n)$,表示从起始点 s 到 n 的最小代价路径与从结点 n 到目标结点的估计最小代价路径的代价之和;且对于搜索图中的每一个结点 n,能满足 $h(n)\leqslant h^*(n)$,则Graphsearch算法即为A*算法.在最坏的情况下,A*算法的时间复杂度是 $O(2^m)$.

A*算法的估价函数 $f(n)=g^*(n)+h(n)$ 中,$h(n)\leqslant h^*(n)$,且 $h(n)$ 愈接近 $h^*(n)$,A*算法的效率愈高.假设对某一问题求解,设有两种估价函数 h_1 和 h_2,且 h_1 和 h_2 都取 h^* 的下界,它们

收稿日期:2001-06-11; **修订日期**:2001-09-07.
作者简介:朱嘉钢(1957-),男,江苏无锡人,工学硕士,讲师.

对应的A*搜索分别记为A_1^*和A_2^*，可以证明以下结论成立：当$h_1 > h_2$时，凡A_1^*搜索到的结点，A_2^*一定搜索到；反之，凡A_2^*搜索到的结点则不一定为A_1^*搜索到. 也就是说，同样是找到一条最佳路径，A_1^*的效率比A_2^*高[1].

2 在A*算法中引入前视机制

为了提高A*算法的搜索效率，探讨在A*算法中引入前视机制.

先引入搜索树的平均搜索宽度的概念，以便进一步分析h值的大小与A*算法效率的关系.

搜索树的平均搜索宽度定义为每向前搜索一步须扩展结点数的平均值. 设搜索树的层数为n，搜索树的平均搜索宽度为q，在得到最优解时被扩展的结点总数为m，则q由下式求出：

$$(1-q^n)/(1-q) = m.$$

式中搜索树的层数对应求解问题的规模，搜索树的平均搜索宽度表明A*算法每向前搜索一步平均需要扩展的结点，引入平均搜索宽度的概念是为了便于对A*算法搜索效率作量化表示和估算. 搜索树的平均搜索宽度越小，则搜索效率越高.

A*算法中，对于取h*下界的估价函数h，h愈大，算法搜索树的平均搜索宽度愈小，算法的搜索效率愈高，算法可求解的实际问题的规模就愈大. 可见h函数的选取对于提高A*算法的效率有很大意义. 但实际上，h函数的选取主要还是凭直觉和经验，要选取理想的h函数是困难的.

为提高h值的质量，提出h函数前视值的概念. 设在搜索树中，n_i是搜索路径上的一个结点，n_j是其直接后裔结点. 对于n_i的所有直接后裔结点n_j，令

$$h(n_i)_1 = \max(h(n_i), \min(K(n_i, n_j) + h(n_j))),$$

其中$K(n_i, n_j)$是结点n_i和n_j之间最佳路径上的实际耗散值，即从n_i到达n_j所付出的代价，则称$h(n_i)_1$是结点n_j的前视1步的h值.

若对于结点n_i的每一个直接后裔结点n_j，先求得n_j的前视1步的h值$h(n_j)_1$，再令

$$h(n_i)_2 = \max(h(n_i), \min(K(n_i, n_j) + h(n_j)_1)),$$

则称$h(n_i)_2$是结点n_j的前视2步的h值.

以此类推，可以定义结点n_i的前视l步的h值$h(n_i)_l$.

根据h的前视值的定义，易证h的前视值必取h*的下界，且前视步数越多，h值的质量越高.

在A*算法中，如果在到达结点n_i时，能够再向前看l步，并得到$h(n_i)_l$，就有$h(n_i)_l \geqslant h(n_i)$. 若在结点$n_i$处用$h(n_i)_l$代替$h(n_i)$作为$n_i$的估价函数值，便可以减小A*算法搜索树的平均搜索宽度，从而提高A*算法的搜索效率，而且随着问题规模的增大，效果愈加明显.

设有两个求解同一问题的A*算法A*1和A*2，其h函数分别为h_1和h_2，$h_1 \geqslant h_2$，计算h_1和h_2所用的时间分别为$t(h_1)$和$t(h_2)$，$t(h_1) \geqslant t(h_2)$，平均搜索树宽分别为q_1和q_2，$q_1 \leqslant q_2$. 由于$t(h_1)$、$t(h_2)$和q_1、q_2是常数，则当问题的规模n满足下式时：

$$t(h_1)(1-q_1^n)/(1-q_1) \leqslant t(h_2)(1-q_2^n)/(1-q_2)$$

A_1^*算法所用的时间开始小于A_2^*算法所用的时间. 两者的时间复杂度分别为$O(q_1^n)$和$O(q_2^n)$. 这说明，在引入了前视机制后，有可能增加A*算法可求解的实际问题的规模.

3 实例仿真结果及分析

选择用于进行仿真的实例是一个派工问题，可描述如下：

设有n个工人、n项工作和一个工作代价表. 工作代价表给出每个工人做每项工作的代价. 假定一个工人可以做3项工作，每项工作有3个工人可以做. 要求分配这n个工人分别做这n项工作，使完成工作的总代价最小. 这一问题需要从$a^n(a>1)$种派法中寻求最佳派法，是一个NP问题.

仿真的方法是：将A*算法和具有前视机制的A*算法用于解决一个派工问题，将问题的规模n分别设置为5、10和15. 在每个规模上，通过随机数发生器产生20组随机数作为模拟数据，每组随机数模拟一个工作代价表中的一组代价. 模拟软件将记录下每种情形下进入Closed表的结点数，计算出每个规模上20个情形下进入Closed表结点数的平均值. 表1给出了仿真结果. 由此求得A*算法的平均树宽为1.370，具有前视机制(前视一步)的A*算法的平均树宽为1.271.

根据平均树宽，可以分别估计出规模较大时两个算法的进入closed表的结点数的平均值. 表2给出了估计结果.

表 1　A* 算法和具有前视机制的 A* 算法用于解决同一个派工问题的仿真结果

Tab.1　Emulation results of a job scheduling problem solved respectively by A* algorithm without and within look-ahead function

问题的规模 n	A* 算法中 20 次仿真进入 closed 表的结点数的平均值	具有前视机制(前视一步)的 A* 算法中 20 次仿真进入 closed 表的结点数的平均值
5	11	9
10	51	35
15	300	109

表 2　规模较大时 A* 算法和具有前视机制的 A* 算法进入 closed 表的结点数平均值的估计

Tab.2　Evaluated mean values of nodes into closed table respectively in A* algorithm without and within look-a-head function under larger scale

问题的规模 M	平均树宽 = 1.370 时进入 closed 表的结点数的平均值	平均树宽 = 1.271 时进入 closed 表的结点数的平均值
20	1 463	443
30	34 149	4 910
40	795 462	54 053
50	18 527 915	594 710

以上估计表明，在 A* 算法中引入了前视机制(前视一步)后，平均树宽减少了 7.2%，在中小规模的情形下，进入 closed 表的结点数减少了一至两个数量级.

可以预计，若这样的前视机制能够前视多步，平均树宽将进一步减少，算法的实用范围将进一步扩大. 同时，仿真表明，由于进入 closed 表结点数的减少，导致进入 open 表的结点数也相应地减少，大大节省了内存空间.

以上仿真采取记录进入 closed 表的结点数的方法，而没有采取记录运行时间的方法，其目的是使仿真结果与仿真环境无关.

4　结　语

作者探讨了在 A* 算法中引入前视机制，以提高 A* 算法的搜索效率. 这一方法，在每扩展一个结点时，以多花一些时间为代价，计算出估价函数的前视值，换取到减小搜索树的平均树宽的好处. 理论和仿真都表明了这一方法的有效性.

参考文献：

[1] 周西苓. 人工智能原理及应用[M]. 北京：航空工业出版社，1997.
[2] 王维平，朱一凡，华雪倩等. 离散事件系统建模与仿真[M]. 长沙：国防科技大学出版社，1997.
[3] 贾纳豫，李慧，楼荣生. 图搜索中 A* 算法的 SQL 解法[J]. 计算机工程(数字化期刊)，1999(8)：10～15.
[4] 曲涧淘，席裕庚，韩兵. 基于进化规划求解 Steiner Tree 问题[J]. 计算机工程(数字化期刊)，1999(8)：23～28.
[5] 邹海明，余祥宣. 计算机算法基础[M]. 武汉：华中理工大学出版社，1993.
[6] 蔡自兴，徐光佑. 人工智能极其应用(第二版)[M]. 北京：清华大学出版社，1997.

(责任编辑：朱　明)