DOI:10.16594/j.cnki.41-1302/g4.2011.08.025

八数码问题的搜索算法比较

欧阳林艳

(福州大学阳光学院计算机工程系,福建福州 350015)

摘 要:搜索策略是人工智能研究的主攻方向之一,采用不同的搜索策略在求解问题的过程中也会存在差异.通过对于八数码的搜索求解分析,采用盲目搜索中的广度优先搜索算法和启发式搜索中的 A* 算法进行实现,将广度优先搜索算法与 A* 算法进行比较,从而评价这两种搜索算法的优劣性.

关键词:搜索策略;广度优先搜索;启发式搜索;A* 算法

中图分类号:TP181 文献标识码:A 文章编号:1009 -4970(2011)08 -0069 -03

0 问题引入

八数码是一种智能游戏,也叫做"九宫格",指的是有 1、2、3、4、5、6、7、8 八个数码和一个空格,当它处于任何一个状态时,空格周围紧临的数码均可以移动到空格位置,最终到达某一目标状态(见下图).

So

2	7	5
3		4
1	6	8

Sg

1	2	3
8		6
7	5	4

上图中初始状态 So 空格紧邻的"3"、"7"、"4"、"6"四个数码则可以选择其中一个移到空格位置,接下来又在新的空格位置选择其紧邻的数码进行移动,以此类推,直到最终到达目标状态 Sg 为止.

八数码的棋盘状态可以是任意的,那么则有 9! =362880 个,因为本身棋盘具有对称的特性,则是 9! /2 = 181440 个,但这个数字还是十分庞大的.如何从如此大的状态空间中寻找到目标状态的解答路径呢?这就是搜索策略应该解决的问题.

1 分析

考虑到由于并非所有初始状态都能到达目标状态,如果每个初始状态都直接用搜索策略进行搜索,则浪费了系统不必要的运算,故对初始状态结点进行判断,如果不能到达目标状态,则无需进行搜索运算,否则才执行.对于判断算法,有个逆序判断法.什么是逆序数呢?在一个排列当中,如果一对数的前后位置与大小顺序相反,即前面的数大于后面的数,那么它们就称为一个逆序.排列中逆序的总数就称为这个排列的逆序数.逆序数为偶数的排列称为偶排列;逆序数为奇数的排列称为奇排列.

如果考虑数码的移动,目前有八个数码,而每个数码都有上、下、左、右四种移动方式,则这样就需要研究 8* 4 =32 种移动规则.故转义理解,将棋盘中数码的移动看成是空格的移动,空格用数字"0"表示,则只需要研究 4 种移动规则即可.

定义 Sij 为矩阵第 i 行 j 列的数码,$Sij \in \{0, 1, \cdots 8\}$,$0 \leqq i, j \leqq 2$,Sij 互不相同;

矩阵(Sij)表示八数码的任何状态.

设空格移动代替数码移动.则至多有四种移动的可能:上、下、左、右.

其中:i_0, j_0 表示空格所在的位置,则 $Si_0j_0 = 0$(0 代表空格)

空格左移规则:

$$\text{if } j_0 \geqq 1 \text{ then } j_0 = j_0 - 1;\ Si_0j_0 = 0$$

解释:如果当前空格不在第一列,则空格左移一位,新的空格位置赋值为 0.

右移规则:$\text{if } j_0 \leqq 1 \text{ then } j_0 = j_0 + 1;\ Si_0j_0 = 0$

上移规则:$\text{if } i_0 \geqq 1 \text{ then } i_0 = i_0 - 1;\ Si_0j_0 = 0$

下移规则:$\text{if } i_0 \leqq 1 \text{ then } i_0 = i_0 + 1;\ Si_0j_0 = 0$

收稿日期:2011 -02 -22

作者简介:欧阳林艳(1982 -),女,湖南衡阳人,助教.

要使任意初始状态的棋盘变为目标状态，则需要调用上述四种规则，调用规则的顺序即是我们需要解决的核心问题，也就是搜索策略问题.

2　搜索策略

搜索策略大致分为两种: 盲目搜索、启发式搜索. 这两种搜索策略的区别就是看是否有在搜索过程中加入启发性知识. 盲目搜索的代表策略有: 广度优先搜索算法、深度优先搜索算法. 启发式搜索代表策略有 A* 算法. 现选择广度优先搜索算法与 A* 算法来进行比较.

2.1　广度优先搜索[1]

从初始节点 S_0 开始，逐层对节点进行扩展并考察它是否为目标节点，在第 n 层的节点没有全部扩展并考察之前，不对第 n + 1 层的节点进行扩展. Open 表中的节点总是按进入的先后顺序排列，先进入的节点排在前面，后进入的排在后面.

搜索过程:

1) 把初始节点 S_0 放入 Open 表.

2) 如果 Open 表为空，则问题无解，退出.

3) 把 Open 表首个节点(即节点 n) 取出放入 Closed 表.

4) 判断节点 n 是否是目标节点，如果是，则求得问题的解，退出.

5) 若节点 n 不可扩展，则转第 2 步.

6) 扩展节点 n，将其子节点放入 Open 表的尾端，并为每一个子节点都配置指向父节点的指针，然后转第 2 步.

2.2　启发式搜索(A* 算法)

A* 算法的核心是一个估价函数: $f(x) = g(x) + h(x)$，它是对从初始状态 S0 经过当前状态 x 到达目标状态 S 的整个路径的估计长度. 其中 $g(x)$ 是对已走距离的估计函数，$h(x)$ 是对当前状态到目标状态距离的估计. 在使用 A* 算法求解问题时，定义的启发函数 h，在满足 A* 的条件下，应尽可能地大一些，使其接近于 h* ，这样才能使得搜索的效率高[2].

搜索过程:

1) 把初始节点 S_0 放入 Open 表，计算 $f(S_0)$

2) 如果 Open 表为空，则问题无解，退出.

3) 把 Open 表首个节点(即节点 n) 取出放入 Closed 表.

4) 判断节点 n 是否是目标节点，如果是，则求得问题的解，退出.

5) 若节点 n 不可扩展，则转第 2 步.

6) 扩展节点 n，用估价函数 $f(x)$ 计算每个子节点的估价值，并为每个子节点配置指向父节点的指针，将其子节点放入 Open 表中，对 Open 表中的全部节点按估价值从小到大进行排序. 然后转第 2 步.

从上述两种算法可以看出，启发式搜索与广度优先搜索的区别就在于是否利用估价函数对 Open 表节点进行排序，不过就是如此区别会让广度优先搜索与启发式搜索在效率上有着很大的不同.

下图是实践过程中广度优先搜索算法与启发式搜索算法的参数比较. 在采用启发式搜索过程中，$g(x)$ 选用的是当前节点已到达的深度，也就是扩展的层数，$h(x)$ 选用的是当前节点在没有任何阻拦的情况下，各个棋牌走到目标状态的位置所用的步数总和(表中初始状态是按每行从左至右顺序排列) .

初始状态	搜索策略	到达目标步数	访问节点数
203 584 716	广度优先搜索	17	10309
	启发式搜索	17	779
458 032 761	广度优先搜索	23	73965
	启发式搜索	23	10619
275 304 168	广度优先搜索	22	61463
	启发式搜索	22	8852
710 862 453	广度优先搜索	18	15945
	启发式搜索	18	1197
382 716 045	广度优先搜索	22	48821
	启发式搜索	22	7281
581 260 743	广度优先搜索	21	44255
	启发式搜索	21	4048

3　结　论

通过实验得到的上表数据可以看出，对于同一可达初始状态，分别采用广度优先搜索与启发式搜索策略(A* 算法) ，他们到达目标所用的步数是一致的，而访问的节点个数差异却很大，采用广度优

先搜索访问的节点数比采用 A* 算法访问的节点数要高出几倍或十几倍.

要在盲目搜索中找到一个解，所需要扩展的节点数目可能是很大的. 因为这些节点的扩展次序是完全随意的，而且没有利用已解决问题的任何特性. 因此，除了那些最简单的问题之外，一般都要占用很多时间或空间(或者两者均有). 这种结果是组合爆炸的一种表现形式[3].

而对于在实验过程中采用的 A* 算法，公式 $f(x)=g(x)+h(x)$ 中 $g(x)$ 与 $h(x)$ 的选择不是没有根据的. 对于 $g(x)$，选择为当前节点的深度，它始终都会比实际解路径上的代价要小或者相等(因为有可能会走弯路)；而对于 $h(x)$，选择在没有任何阻拦的情况下各棋牌走到目标状态位置的总和，这个始终也会比实际路径要小或相等，因为几乎不可能每个棋牌走到自己的位置都没有任何障碍物. 对于 A* 算法来说，让 $h(x)$ 小于或等于实际代价，目的是保证 A* 算法能获得最优解.

搜索算法的灵魂在于搜索策略，而搜索策略的主要任务是确定如何选取规则的方式. [4] 广度优先算法和 A* 算法虽然都能找到最优解，但是从上面数据可以看出，对于同一初始状态，A* 算法扩展的节点数远远小于广度优先搜索算法扩展的节点数. 可见，采用 A* 算法搜索策略要比广度优先搜索算法更优.

参考文献

[1] 王永庆. 人工智能原理与方法 [M]. 西安: 西安交通大学出版社，1998.

[2] 马少平，朱小燕. 人工智能 [M]. 北京: 清华大学出版社，2004.

[3] 蔡自兴，徐光祐. 人工智能及其应用(第三版) [M]. 北京: 清华大学出版社，2003.

[4] 詹志辉，胡晓敏，张军. 通过八数码问题比较搜索算法的性能 [J]. 计算机工程与设计，2007，(6).

[责任编辑 胡廷锋]

Searching Algorithms Comparison of the Eight Digital Problem

OUYANG Lin-yan

(Computer Engineering Department, Sunshine College of Fuzhou University, Fuzhou 350015, China)

Abstract: Search strategy is one of the main directions in the artificial intelligence research. Using different search strategies in the process of solving problem also may have differences. Through the analysis of the search solution for eight digital, the Breadth-first search algorithm, a kind of the blind searching, and A * algorithm, which is one of the heuristic search, are used to implement it. Breadth-first search algorithm and A * algorithm are compared to evaluate the superiority between these two search algorithms.

Key words: search strategy; breadth-first search; heuristic search; A* algorithm