

深度优先搜索算法及其改进

龚建华

(解放军通信指挥学院 湖北 武汉 430010)

摘 要:对于一些简单的搜索问题或者不便构建启发式搜索算法的问题,深度优先搜索算法常是解决问题的有效办法。首先对深度优先搜索算法的基本原理进行描述,在此基础上分析深度优先搜索算法的不足之处,最后对深度优先搜索算法进行改进,并将改进的深度优先搜索算法应用于农夫过河问题,得到2个可行的解。

关键词:深度优先搜索;启发式搜索;农夫过河;栈

中图分类号:TP18

文献标识码:B

文章编号:1004-373X(2007)22-090-03

Depth Priority Algorithm and Its Improvement

GONG Jianhua

(Commanding Communication Academy of PLA, Wuhan, 430010, China)

Abstract: To some simple search problems or some problems that are difficult to be built with heuristic search algorithm, the depth priority search algorithm is always an effective method to solve problems. At First, the basic theory of depth priority search algorithm is introduced. On the basis of it, defects of this algorithm are analyzed. At last, improvement of depth priority search algorithm is described and applied to passing river problem. With the improved algorithm, two results of passing river problem are obtained.

Keywords: depth priority search; heuristic search; passing river problem; stack

1 引言

搜索技术是人工智能的一个重要研究内容,对于一个问题的求解过程,实质上是从开始状态,利用规则进行状态的改变,一直到达目标状态,把状态的改变连接起来就成了搜索路径,现在的目的就是要找到这条从起始状态到达目标状态的路径。

搜索算法可分为盲目搜索算法和启发式搜索算法2种,二者的区别在于,启发式搜索算法的每一步都试图向着目标状态方向进行搜索,而盲目搜索算法则是每一步按照固定的规则进行搜索,然后判断是否达到目标状态,相比之下启发式搜索算法克服了盲目搜索算法的盲目性问题。

虽然启发式搜索算法可以解决盲目搜索算法的盲搜索问题,但是,在许多实际问题求解中,缺少一些必要的信息来构建启发式搜索算法,此时采用盲目搜索算法仍然是解决问题的有效手段。盲目搜索算法有2种,一种按照广度优先展开搜索树的搜索算法,叫广度优先搜索法,一种按照深度优先展开搜索的搜索算法,叫深度优先搜索法。本文将介绍深度优先搜索算法,然后介绍该算法的改进。

2 深度优先搜索算法及其分析

2.1 深度优先搜索算法

深度优先搜索算法是把最近刚产生的结点优先扩展,直到达到一定的深度限制。若未找到目标或无法再扩展时,再回溯到另一个结点继续扩展。

从初始状态S开始,利用规则生成搜索树下一层任一结点,检查是否出现目标状态G,若未出现,以此状态利用规则生成再下一层任一结点,再检查是否为目标状态G,若不是,继续以上操作过程,一直进行到叶结点(即不能再生成新状态结点),当没有发现目标状态G时,回溯到上一层结果,取另一可能扩展搜索的分支,生成新状态结点;若仍不是目标状态,就按该分支一直扩展到叶结点,若仍没有发现目标状态,采用相同的回溯办法回退到上层结点,扩展可能的分支生成新状态;如此一直进行下去,直到找到目标状态G为止。搜索过程如图1所示,图1中箭头上下的标号表示搜索的顺序,从图中可以看出,经过17步,从初始状态S搜索到了目标状态G。

深度优先搜索算法描述为:首先定义数据结构Open栈和Close表,其中:Open栈,先进后出,存放待扩展的结点;Close表,存放已被扩展过的结点。

算法操作步骤为:

Step1 把起始结点S先放到Open栈中,

Step2 如果 Open 栈为空,则失败退出,否则继续;

Step3 从 Open 栈中取最前面的结点 node 移到 Close 表中;

Step4 若 node 结点是叶结点(若没有后继结点),则转向 Step2;

Step5 扩展 node 的后继结点,产生全部后继结点,并把压入 Open 栈。各后继结点的指针指向 node 结点;

Step6 若后继结点中某一个为目标结点,则找到一个解,成功退出;否则转向 Step2 循环。

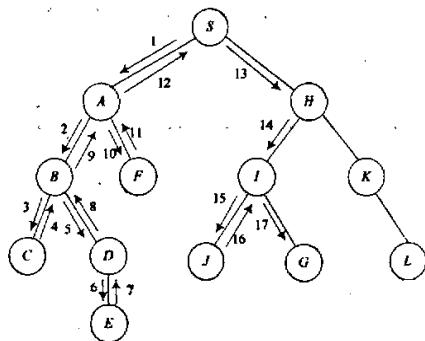


图1 深度优先搜索过程示意图

2.2 深度优先搜索算法分析

(1) 算法中没有描述考虑到在生成的下一层结点中,有可能会出现的上一层或上上一层的结点,这种情况在深度优先搜索算法中容易导致搜索进入死循环。在图1中,如果状态C等于状态A,即C结点可以再扩展B结点,由于B又可以扩展C结点,这样,在深度优先搜索过程,在这一分支上容易进入 $S \rightarrow A \rightarrow B \rightarrow C \rightarrow B \rightarrow C \rightarrow$ 的死循环。

(2) 算法中存在过于盲目性的问题,第一次出现某个无效结点(从该结点开始一直搜索下去,到达不了目标结点),以后再次出现这个无效结点时没有跳过该无效结点。

(3) 算法第6步找到一个解后就退出,这样会漏掉其他的解,应当继续搜索下去,直到搜索结束或者人为干预停止搜索为止,尽可能将所有的解搜索出来。

3 深度优先搜索算法的改进

针对深度优先搜索算法的缺点,对其进行必要的改进,主要是针对死循环和无效结点的处理,并一直搜索下去,直到搜索全部解。

3.1 创建 DisabledNodes 动态数组和 PassedNodes 动态数组

首先创建一个公共的 DisabledNodes 动态数组,主要记录:所有的不符合规则的状态结点;沿状态结点所有子状态结点方向搜索下去,一直都找不到目标结点的状态结点。

然后为所有的状态结点创建 PassedNodes 动态数组,记录从初始状态结点开始,一直到该状态结点为止所经过

的所有状态结点。

3.2 构建递归搜索函数 DepthSearch

深度优先搜索算法比较容易通过递归搜索的方法来实现,设当前状态结点为 Node, Node 扩展结点数组为 SubNodes,定义一个递归搜索函数 DepthSearch(Node, Node 的 PassedNodes),具体描述如下:

```
DepthSearch(Node, Node 的 PassedNodes)
开始
扩展 Node 得到 SubNodes 数组
For I=1 to SubNodes 的总数
SubNodes[I] 的 PassedNodes 数组 = PassedNodes + SubNodes[I]
EndFor
初始化 Output=False 且 DeepOutput=False
For I=1 to SubNodes 的总数
If SubNodes[I] = G
输出 SubNodes[I] 的 PassedNodes 数组,并且 Output=True
ELSEIF SubNodes[I] 不在 DisabledNodes 数组或 Node 的 PassedNodes 数组中
DeepOutput = DepthSearch(SubNodes[I], SubNodes[I] 的 PassedNodes 数组)
EndFor
Result = Output or DeepOutput
IF Result = False 则将 Node 加入 DisabledNodes 中
返回 Result
结束
```

4 改进的深度优先搜索算法应用

4.1 农夫过河问题描述

一个农夫,带着一只狼、一只羊和一棵白菜要过河,河边有一条船,过河有以下规则:

- (1) 农夫一次最多能带一样东西(或者是狼、或者是羊、或者是白菜)过河;
- (2) 当农夫不在场时狼会吃掉羊;
- (3) 当农夫不在场时羊会吃掉白菜。

现在要求为农夫想一个方案,能将3样东西顺利地带过河。从初始状态开始,农夫将羊带过河,然后农夫将羊又带回来也是合规规则的,然后农夫又将羊带过河仍然是合规规则的,如此这般,搜索过程便进入了死循环,因此,在这里,采用改进的搜索算法进行搜索。

4.2 农夫过河问题搜索结果

在任何状态下,可以根据农夫过河规则(1)来扩展其子状态,并根据规则(2)和规则(3)来判断子状态的合法性。如果用4个比特从左至右分别表示农夫、狼、羊和白菜,则状态1111表示农夫、狼、羊和白菜都在河边1,状态0101表示农夫将羊带到了河边2,很显然0011,0110,1100,1001,0111,1000这些状态因为不符合规则(2)和规则(3)而被设置为禁止状态。

采用改进的深度搜索算法可以搜索出2个解,分别如表1和表2所示。

表1 农夫过河问题解一

步骤	河边1	过河方式	河边2
1	农夫、狼、羊、白菜	农夫带羊过河	
2	狼、白菜	农夫独自回来	农夫、羊
3	农夫、狼、白菜	农夫带白菜过河	羊
4	狼	农夫带羊回来	农夫、羊、白菜
5	农夫、狼、羊	农夫带狼过河	白菜
6	羊	农夫独自回来	农夫、狼、白菜
7	农夫、羊	农夫带羊过河	狼、白菜
农夫、狼、羊、白菜			

表2 农夫过河问题解二

步骤	河边1	过河方式	河边2
1	农夫、狼、羊、白菜	农夫带羊过河	
2	狼、白菜	农夫独自回来	农夫、羊
3	农夫、狼、白菜	农夫带狼过河	羊
4	白菜	农夫带羊回来	农夫、狼、羊
5	农夫、羊、白菜	农夫带白菜过河	狼
6	羊	农夫独自回来	农夫、狼、白菜
7	农夫、羊	农夫带羊过河	狼、白菜
农夫、狼、羊、白菜			

5 结语

经过农夫过河问题的验证,表明经过改进的深度优先搜索可以避免搜索进入死循环,还可避免重复无效的搜索,同时能够尽可能地搜索出问题所有解。对于不便构建启发式搜索的问题,改进的深度优先搜索将具有十分广泛的应用。

参考文献

[1] 金聪,戴上平,郭京雷,等.人工智能教程[M].北京:清华大

学出版社,2007.

- [2] 陈文伟.决策支持系统教程[M].北京:清华大学出版社,2004.
- [3] 吴泉源,刘江宁.人工智能与专家系统[M].长沙:国防科技大学出版社,2000.
- [4] 赵丹青,胡宝玉.状态空间的几种搜索算法讨论[J].中南民族学院学报:自然科学版,2001,20(3):46-49.
- [5] 雷伟刚.城市地下管网信息系统中管网追踪算法[J].同济大学学报,2003,30(1):99-103.
- [6] 赵广,王保平,刘道华.深度优先的公式发现算法[J].中州大学学报,2003,20(4):100-101.

作者简介 龚建华 男,1973年出生,湖北麻城人,解放军通信指挥学院讲师。主要从事人工智能决策支持系统研究与开发。

(上接第89页)

用者,下层通过接口向上层提供服务,服务的使用者不必知道服务的提供者是如何实现的,那么下层的修改不会影响上层,使系统能够快速响应需求的变更,提高了系统的维护性和扩展性。该体系结构是分布式的,分布式组件可以运行在多个应用服务器上,各服务器通过集群而协同工作,具备负载均衡和透明的失败转发机制,提高了系统的可靠性和伸缩性。该体系结构融合了J2EE标准使得开发出的产品不仅可以跨硬件、跨操作系统,还可以跨应用服务器,这些都大大提高了投资的回报率。实际运行证明,本文提出的技术方案具有可行性和先进性,在一定程度上解决了当前远程教育平台体系结构设计的不足,达到系统设计的目标。

参考文献

[1] Erich Gamma, Richard Helm, Ralph Johnson, et al. Design Patterns: Elements of Reusable Object-Oriented Software

[M]. Addison-Wesley, 1995.

- [2] 孙新,冯霞,郝成义.国内外远程教育系统分类与比较[EB/OL]. www. http://open.edu.cn/ycjy/.
- [3] 胡晓玲,杨改学.远程教育发展趋势探讨[J].中国远程教育,2001(5):78-79.
- [4] 武法提.网络教育应用[M].北京:高等教育出版社,2003.
- [5] 袁梅冷,黄烟波. J2EE应用模型中MVC软件体系结构的研究与应用[J].计算机应用研究,2003,20(3):147-149.
- [6] Roman E D. Master Enterprise JavaBeans[M]. Second Edition. Wiley & Sons, Inc, 2002.
- [7] 房丙午.基于J2EE架构的远程教育平台的设计与实现[D].北京:中国科学技术大学,2005.
- [8] Deepak Alur, John Crupi, Dan Malks, Core J2EE Patterns: Best Practices and Design Strategies[M]. Prentice Hall, PTR, 2001.
- [9] 曹春萍,王志民. MVC设计模式的研究及其应用[J].现代电子技术,2005,28(20):80-82.

作者简介 房丙午 男,1974年出生,安徽枞阳人,硕士,工程师。主要研究方向为软件工程与网络分布式计算。

张佑生 男,1941年出生,湖南浏阳人,教授,博导。研究方向为计算机图形学,CAD和智能CAD,人工智能。