

迷宫算法的改进与动态实现

胡湘萍, 陈利军

(河南经贸职业学院 信息管理系, 河南 郑州 450053)

摘要: 本文在对广度优先迷宫搜索算法和深度优先迷宫搜索算法进行了仔细比较与探讨之后, 提出一种新的算法: 目标优先法。即每次向下一个位置搜索时, 按当前位置的各方向靠近目标点的距离去选择方向。使得搜索过程在较短时间内能够快速从入口向出口目标逼近。然后从数据输入输出, 程序设计等方面讲述了这种带优先级的算法的实现。并将此算法用 Java 语言在 JDK 上实现其搜索过程的画面, 模拟其算法实现过程。最后, 将此算法与传统的广度优先和深度优先算法优缺点进行了综合比较。

关键词: 迷宫算法; 广度优先; 深度优先; 目标优先; 堆栈

中图分类号: TP312 文献标识码: A 文章编号: 1009-3044(2007)08-20490-02

Improvement and Realization of Labyrinth Algorithm

HU Xiang-ping, CHEN Li-jun

(1. Administration of Information Department, Henan Economy and Trade Vocational College, Zhengzhou 450053, China)

Abstract: The thesis has compared and analysed two kinds of algorithms to maze problem, the width first search algorithm and the depth first search algorithm. Then it provides a new kind of algorithm, target first search algorithm. It means when searching the next gridding, it will search in the direction whose abuttal gridding is nearest to the target gridding. The search algorithm could find a way from the entrance to the exit very fast. And the thesis has expounded the realization process from several aspects detailed, such as data in data out, program design and so on. And it has been designed into java program and has been performed on SDK with dynamic graph which simulates the new search algorithm. At last, the thesis has compared the advantages and disadvantages of the three maze algorithms.

Key words: maze search algorithm; width first search; depth first search; target first search; stack

1 引言

在学习《数据结构》的时候, 我们遇到一类迷宫问题, 一般采用如同“波浪扩散式”的广度优先搜索或是不断“回溯”的深度优先搜索算法搜索出一条路径。这两种方法各有利弊。用前者在有解的情况下一定能得到一条最短路径但却要访问到所有的点, 浪费了较多搜索时间; 后者不断回溯, 直到找到一条路径, 实现简单, 得到的不一定是最短路径, 在特殊情况, 也有可能浪费很多搜索时间。

目前, 迷宫问题已广泛应用于机器人行走, 电路布线等方面。由于其灵活性, 所以其扩展方向应增加到 8 个。我们希望通过回溯思想和堆栈思想改进这两种搜索算法, 创立一个搜索速度更快的算法——目标优先算法。然后用 Java 语言在 SDK 上实现这个算法: 通过生成一个图形画面的迷宫, 然后用改进的算法控制搜索过程, 逐步显示路径, 从而使整个搜索过程清晰。

2 问题抽象模型

迷宫是一个存在障碍物和通道的矩形区域。在此区域内, 对象要通过不断的尝试, 找到从入口到出口的路径。最早的迷宫仅有四个方向, 但后来已延伸到八个方向。计算机中用二维数组表示迷宫, 如图 1 所示, $maze[6][8]$ 表示一个 6 行 8 列的迷宫, 数组中每个分量 $maze[i][j]$ 的值或为 0 或为 1, 前者表示可以走通, 后者表示遇到障碍物。不失一般性, 可设迷宫入口的坐标为 $[0][0]$, 出口的坐标为 $[m-1][n-1]$, 且设 $maze[0][0]=0$ 和 $maze[m-1][n-1]=0$ 。



图 1 迷宫及其最短路径

3 目标优先法

3.1 深度优先算法基本思想

收稿日期: 2007-04-16

作者简介: 胡湘萍 (1982-), 女, 河南经贸职业学院信息管理系助教, 解放军信息工程大学在读研究生。

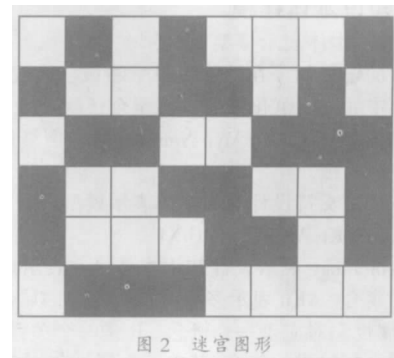


图 2 迷宫图形

首先, 在计算机中可以用如图 2 表示迷宫。图中的每个方块或为通道 (以空白方块表示), 或为墙 (以黑方块表示)。所求路径必须是简单路径, 即在求得的路径上不能重复出现同一通道块。

假设“当前位置”指的是“在搜索过程中某一时刻所在图中某个方块位置”, 则求迷宫中一条路径的算法的基本思想是: 若当前位置“可通”, 则纳入“当前路径”, 并继续朝“下一位置”探索, 即切换“下一位置”为“当前位置”, 如此重复直至到达出口; 若当前位置“不可通”, 则应顺着“来向”退回到“前一通道块”, 然后朝着除“来向”之外的其他方向继续探索; 若该通道块的四周四个方块均“不可通”, 则应从“当前路径”上删除该通道块。所谓“下一位置”指的是“当前位置”四周八个方向 (东, 东南, 南, 西南, 西, 西北, 北, 东北) 上相邻的方块。

假设以栈 S 记录“当前路径”, 则栈顶中存放的是“当前路径上最后一个通道块”。由此, “纳入路径”的操作即为“当前位置入栈”; “从当前路径上删除前一通道块”的操作即为“出栈”。

3.2 目标优先算法创新思想

从“当前位置”向“下一位置”搜索的过程中, 并不是直接按照顺时针方向去搜索下一邻接点。而是要计算其周围相邻 8 个邻接点的优先级。按优先级从大到小的顺序依次搜索八个方向。因为, 优先级越大, 说明该点越趋向于目标点, 所以应优先搜索。那么各方向的优先级如何确定呢? 我们确定的优先级的函数如下:

```
void Prior(PosType e){ }
```

//为结点 e 的 div[8]和 dirnum[8][2]赋值

div[8]是一个数组,记录当前结点 e 的各个方向的优先级情况,div[i]值为对应的优先级。i = 1, 2, 3, 4, 5, 6, 7, 8 分别代表东,东南,南,西南,西,西北,北,东北方向,即顺时针排列。此函数给 e. div[8][2]赋值;dirnum[8][2]是一个二维数组,在计算当“前点位置”的方向时,其意义是:第一列存放按优先级从大到小排列的方向数字序号,第二列初始值全为 0,当该方向被访问了,则将该值记为 1,表示已被访问过。此函数给 dirnum[8][2]赋值,数据来源于 div[8]。

Div[i]的值计算方法如下:

设 i 方向的坐标为:(Xi, Yi), 目标的坐标为:(Xt, Yt), 则:div[i]= $(M+N) - (\sqrt{2} \cdot \min(|Yt - Yi|, |Xt - Xi|) + |Yt - Ys| - |Xt - Xs|)$ 。???

公式几何意义为:div[i]等于(M+N)减去当前方向点与目标点之间的网格数。这种数网格的方法类似曼哈顿距离,比欧式距离的计算明显少一个数量级。避免了大量浮点数的运算,计算简便。优先级数值越大,说明 i 方向点越趋向于目标点。于是应优先考虑,使当前点向趋于目标单元的方向进行搜索。

两点之间的网格数计算公式几何解释如下图[2]:

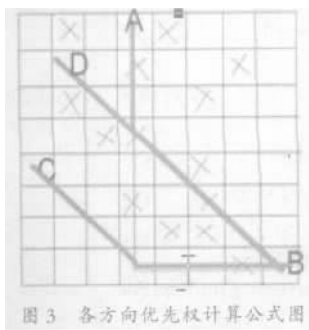


图3 各方向优先级计算公式图

如图:

A 与 B 之间的网格数为 A 到 B 的红线经过的网格数,为 5+3=8;

D 与 B 之间的网格数为 D 到 B 的线线经过的网格数,为 7;

C 与 B 之间的网格数为 C 到 B 的红线经过的网格数,为 4+4=8;

当然,当方向点与目标单元之间有障碍时,此距离会更大,但这只是粗略估计。可以照顾到大多数方向点,使得当前结点能够以较快的速度逼近目标结点,并且可以有效的减少大量的扩展结点,使得搜索速度大大提高。

4 数据的输入输出与迷宫模型

迷宫最短路径问题的数据输入主要包括程序规模、数字化迷宫形成、行进方向设定等。其数据输出包括迷宫图形和运行过程及结果。

4.1 输入迷宫问题的大小规模

采用 m 行 n 列数值矩阵描述迷宫,控制问题规模的系数 m、n 都应该是正整数,反映迷宫的长宽尺度。Pat(路墙比)也是一个正整数,用来调整迷宫中 0 与 1 的个数比例。该数越大,迷宫中 0 的个数越多,宫墙相对越少,迷宫越容易通过。4.2 建立数值迷宫图形可用一个二维数组 maze[m+2][n+2]模拟该数值迷宫。其中 maze[0][j],maze[i][0],maze[m+1][j]和 maze[i][n+1]均被赋值为 1。数组元素为 0 表示该位置允许通过,数组元素为 1 表示该位置已被封锁。maze[1][1]和 maze[m][n]分别为迷宫的入口和出口。这样便得到了以矩阵形式排列的迷宫数值模型。其余各点初始化时,均被赋值为 1。在设置迷宫时,除了四周固定用 1 表示外,其余各点由用户随机生成。

4.3 坐标关系

$$\begin{cases} g = d[i] \\ h = -d[i] \end{cases} \quad v = 0, 1, 2, \dots, 7$$

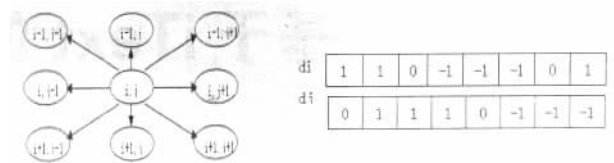


图4 方向关系图

4.4 数据输出

首先输出模拟迷宫的二维数组,若其中存在最短路径,则由出口回溯到入口,通过一个栈暂存路径,最后打印这一条路径,如下所示:

(1,1),(i,j), ..., (m,n)

同时在迷宫画面上显示路径行走过程,鼠标每点一次,显示一格,只到所有路径显示出来。若无通路,则打印:“未找到路径!”

4.5 搜索过程界面设计及演示效果

根据迷宫矩阵的数据,首先生成初始的迷宫界面(M 行 N 列的矩形画面)。初始时,任意方块均为黑色(表示不可通),需用鼠标随机点击画面上的方格,来设置迷宫。每点一格,表示该方格被激活,成为可通结点(显示绿色)。根据目标优先思想,设计搜索函数,最后在 SDK 上运行。鼠标每点击一下,则从入口到出口走一步,最后显示出完整路径。

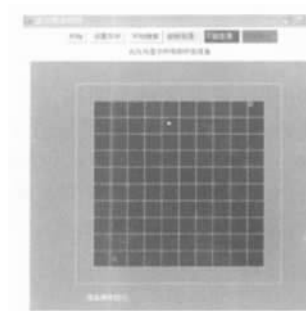


图5 初始界面设计

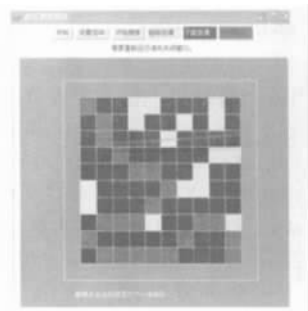


图6 运行过程中显示路径画面

5 核心算法

坐标点类型定义略,主要元素结构类型定义如下:

```
typedef struct{
    int ord;          //通道块在路径上的“序号”
    PosType seat;     //通道块在迷宫中的“坐标位置”
    int di;           //从此通道块走向下一通道块的“方向”
    int div[8];       //八个方向的优先级
    int dirnum[8][2];
    //按优先顺序排列的各个方向标号和访问情况
}SElenType;          //栈的元素类型

Status MazePath(MazeType maze, PosType start, PosType end){
    /* 若迷宫 maze 中存在从入口 entrance 到出口 exit 的通道,则
    求得一条存放在栈中(从栈底到栈顶)的路径,并返回 TRUE; 否则
    返回 FALSE */
    InitStack(S);
    curpos=start;    //设定“当前位置”为“入口位置”
    curstep=1;       //探索第一步
    do{
        if(pass(curpos)){
            //当前位置可通过,即是未曾走过的通道块
            FootPrint(curpos); //留下足迹
            e = (curstep, curpos, 1);
            push(S, e);        //加入路径
            if(curpos==end)return (TRUE); //到达终点(出口)
            prior(curpos);     //计算优先级
        }
    }while(1);
```

(下转第 576 页)

程中实施了以下三个步骤:

2.3.1 理解基本概念和基本操作

每一章节都制定相应教学目标和要求。每次都是三节多媒体教室理论课,运用 PowerPoint 制作多媒体教学课件,用鼠标菜单操作,利用投影演示。适时配上传统的板书,可更加直观且生动的讲解教学内容。对于概念抽象的,巧用生活中的实例,进行形象化比喻。如在讲解文件目录结构这一节内容时,用一棵树的枝叶结构来比喻文件目录结构,并用图形加以解释,这样学生理解起来就相对容易些。又如在讲解数制转换内容时,很多学生难以理解和掌握二、十进制转换表,用“排队”方式形象讲解,把速记的学习技巧传给学生。若是学生操作中的问题,在上机课内加以指导,鼓励学生分析产生错误的原因并找到解决方法。如利用 Excel 制作工资表,如何运用公式函数计算职工的应发工资、实发工资和合计等等。

2.3.2 注重上机实践

计算机基础课的教学成果很大部分体现在学生实践技能的提高上。学生会做,并不等于熟练,为了提高熟练程度,严格要求学生必须在规定时间内完成每一道习题,让学生多上机操作,其目的是从培养学生的操作技能入手,让学生多动手、多动脑,提高操作的准确性、迅速性、灵活性和协调性。同时,还向学生适当介绍一些操作技巧。如:用 Word 的硬换页、软回车,创建目录,自动翻页和产生摘要等等。

2.3.3 改变考试模式

我将考核成绩分为三个部分:平时作业及设计、上机考试、期末考试。其比例设为 20%、40%和 40%。平时作业设计主要是为了检查学生复习和巩固所学的基本内容,加深对基本理论和知识的理解。期末考试全面综合考核学生的知识掌握程度和能力水平。计算机基础课程面广量大(每学年有 3 千人次),无纸化考试与传统考试方式相比,其效率的提高将是十分显著的。并且对推进教育改革,提高课程建设水平、加快教学现代化进程,将产生积极影响。也让学生进一步适应合理、科学的考试模式。

3 变鼓励为动力

即是注重第二课堂教学,计算机基础知识的学习和应用技能的提高,单凭课堂教学是远远不够的,因此,我们每年可举办全院性的计算机基础知识和操作技能大赛。通过大赛,极大地激发了学生学习计算机的兴趣,主要强化实践技能,用大量的操作技巧来提高提高操作速度,同时也深化了学生对日常学习内容理解,增强学生学习后的成就感。还可利用计算机等级考试、微软等相关认证,增强学生就业砝码等,提倡“双证”教育,鼓励学生主动、努力学习。

4 旧经验启发新创意

计算机的方便实用性人人皆知。经常借助计算机处理很多我们工作和生活中的问题。但计算机系统的稳定和安全性是现在更关注的,一旦运行不稳定或是系统崩溃,将造成不可估量的损失。在教学过程中,我要求学生在处理文件的过程中,要注意经常保存,以及强调对一些重要的数据做好备份是非常重要的。同时逐步地给学生讲授一些应急措施、鼓励学生订阅一些报刊、杂志,从而积累更多的软硬件计算机维护方面的知识。

5 总结

以上是我在计算机基础课程教学中一些尝试性的做法。在先进的教育科学理论指导下,充分发挥计算机、多媒体和网络等现代化教育技术手段作用。实践检验一切,还有更多、更好的教学方法需要我经过不断的试验、探究、总结、提高。

参考文献:

- [1]教育部人事司.高等教育学[M].北京:高等教育出版社 2002.
- [2]伍新春.高等教育心理学[M].北京:高等教育出版社 2002.
- [3]李春秋.高校教师职业道德修养[M].北京:北京师范大学出版社,2000.
- [4]中国高等教育[M],2001(15-23).
- [5]钱开益,楼诗风.计算机基础及应用[J].合肥:万博学院,2001.
- [6]吴国凤.计算机文化基础[M].合肥:安徽大学出版社,2000.

(上接第 491 页)

```
curpos=NextPos(curpos, e.dir[8], e.dirnum[8][2]);
//下一位置是当前位置方向中优先级最小的方向单元
curstep++;           //探索下一步
} //if
else{                 //当前位置不能通过
if(! StackEmpty(S)){
Pop(S, e);
While(e 的八个方向都被访问过 &&! StackEmpty(S)){
MarkPrint(e.seat);Pop(S, e);
//留下不能通过的标记,并退回一步
} //while
if(e 的八个方向还有未被访问过的点)
重新按优先级选择下一个方向;Push(S, e);
//换下一个方向探索
curpos=NextPos(e.seat, e.dir[8], e.dirnum[8][2]);
//设定当前位置是该新方向上的相邻块
} //if
} //if
} //else
} while(! StackEmpty(S));
return(FALSE);
} //MazePath
```

7 算法比较

由表 1 可见,当迷宫规模比较小的时候,三种方法的差别不

是很明显,特别是深度优先和目标优先,在找到正确路径时,要访问的结点数几乎相同。但是随着规模不断增大,目标优先法的优势就更明显了。因为,目标优先的原理,就是以与目标点的网格距离远近为优先级(越近越优先),使得当前点不断的向目标点靠近。特别的,当多次生成随机迷宫,进行多次测试的时候就更易显示这种优越性。

表 1 三种搜索算法综合比较

比较 项目 实验 次数	规模 (m*n)	路墙比 (pat)	扩展结点数			劳动比		
			广度 优先 (g)	深度 优先 (s)	目标 优先 (k)	广度 优先	深度 优先	目标 优先
S[1]	3*3	5/4	4	3	3	0.44	0.33	0.33
S[2]	5*5	12/13	11	7	7	0.44	0.28	0.28
S[3]	6*8	25/23	29	9	8	0.61	0.19	0.17
S[4]	10*10	36/64	30	29	228	0.30	0.29	0.25

参考文献:

- [1]孙巧榆,潘荫荣,胡幼华,孙强.八方向走迷宫算法[M].计算机工程,2004,30(1).
- [2]Hoel J H.Some Variations of Lee' s Algorithm.IEEE Trans on Electron Computer,1976,25(1):19-24.
- [3]Deo N,Pang Chiyin.Shortest - path Algorithms:Taxonomy and Anno - tation.Networks,1984,14:275-323.