Path Planning Algorithm of Maze based on Ant Colony Optimization

Zhang Xiaohong [1,2]
[1,]School of Communicaion and Information Eengineering,
Xi'an University of Science and Technology
xi'an City, China
zxh3311_cn@xust.edu.cn

He Bo [2],Niu Xiao [1]
[2,] School of Information and Control Engineering,
Xi'an University of Architecure and Ttechnology
Xi'an City, China

***Abstract*—Searching optimal path is an important function of the maze question. For the shortage of the traditional algorithm to solve the maze problem, this paper at first analyzes the charactaristic of Ant Colony Optimization(ACO) and designs the algrithm of dynamic path planning based on ACO. Secondly, after running the programs coded c#, this paper records the time complexity, space complexity and operating results of the algorithm based on ACO and the Dijastra algorithm. As the conclusion,this paper compares them qualitatively and quantitativly, and describes the next work in future.**

Keywords—maze;ACO; path planning; time complexsition; space complexsition

I. 引言

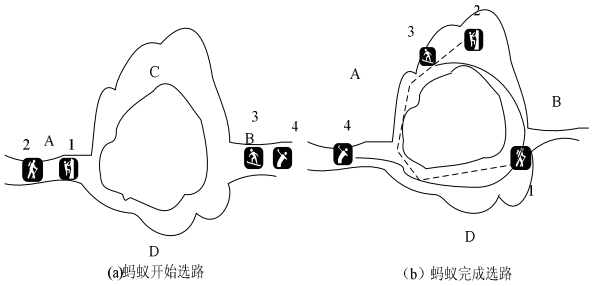
迷宫问题来源于心理学的一个古典实验。在该实验中，把一只老鼠从一个无顶大盒子的门放入，在盒中设置了许多墙，对行进方向形成了多处阻挡。盒子仅有一个出口，在出口处放置一块奶酪，吸引老鼠在迷宫中寻找道路以到达出口。

迷宫问题存在着很多解决方法,如图、遗传算法、粒子群算法等。其中 Dijakstra 算法能得出最短路径的最优解，但由于它遍历计算的节点很多，所以效率低。文献[4]针对遗传算法运算速度低、容易陷入局部最优值、早熟收敛等缺点,对标准遗传算法进行了改进和优化。文献[6]用粒子群算法解决传统的迷宫问题,用求解次优可行解代替最优解来节省求解时间。

II. 蚁群算法原理

蚁群算法是 M.Dorigo 等人利用蚁群搜索食物的过程中蚂蚁个体之间通过间接通讯与相互协作最终达到目的的特性，通过人工模拟蚁群搜索食物的行为来求解 TSP 问而得出的一种智能优化算法。

人们经过大量研究发现，蚂蚁个体之间是通过一种称之为外激素(pheromone)的物质进行信息传递，从而能相互协作，完成复杂的任务。蚂蚁在运动过程中，能够在它所经过的路径上留下这种物质，而且能够感知这种物质的存在及其强度，并以此指导自己的运动方向。蚂蚁倾向于朝着该物质强度高的方向移动。因此，由大量蚂蚁组成的蚁群的集体行为便表现出一种信息正反馈现象：某一路径上走过的蚂蚁越多 ，则后来者选择该路径的概率就越大。蚁群正是通过这种信息的交流最终找到食物源与蚁巢之间的最短路径。如图 1 (a)所示，在网络的某两个节点 A 与节点 B 之间有两个支路 ACB 和 ADB，其中 ACB 支路的长度大于 ADB 支路的长度;假设在节点 A、B 各有两只蚂蚁，蚂蚁 1、2 由节点 A 向节点 B 行进，而同时蚂蚁 3，4 由节点 B 向节点 A 行进。



(a)蚂蚁开始选路　　(b）蚂蚁完成选路

图1 蚁群算法模拟图

在初始情况下，由于各个支路上没有任何信息素的痕迹存在，所以在节点 A 和节点 B 处蚂蚁选择两条支路的概率是均等的;于是蚂蚁 1 和 2 将分别沿着支路 ACB 和 ADB 向着节点 B 行进，同样蚂蚁 3 和 4 也将分别沿着支路 BCA 和 BDA 向着节点 A 行进 ，由于蚂蚁行进的速度相同，在一段时间之后，蚂蚁 2 和蚂蚁 4 经过支路 ADB 分别到达节点 B 和 A，而蚂蚁 1 和 3 却还在支路 ACB 的途中(图 1(b))。在蚂蚁行进的过程中每个蚂蚁均留下了同样数量的信息素的痕迹。这时可以看出，支路 ADB 上留下的信息素的痕迹浓度要高于支路 ACB 上的信息素浓度；此后若再有蚂蚁到达节点 A 和 B 时，由于受到信息素痕迹的诱导它们选择支路 ADB 的概率就会较大，反过来它们又不断地增加支路 ADB 上的信息素痕迹的浓度，形成正反馈作用；与此同时，遗留在支路 ACB 上信息素的痕迹还会因不断的挥发而进一步的减弱。这样一来，选择走支路 ADB 的蚂蚁就会越来越多，选择走支路 ACB 的

978-1-4244- 7618-3 /10/$26.00 ©2010 IEEE

蚂蚁就会越来越少，最后呈现有较强的信息素痕迹的那些支路便会形成一条从蚁穴到食物源的最短路径。

III. 基于蚁群算法的迷宫路径规划算法设计

本文在设计迷宫时，首先，要让蚂蚁能够避开障碍物，就必须根据适当的地形给它编进指令让他们能够巧妙的避开障碍物，其次，要让蚂蚁找到食物，就需要让他们遍历空间上的所有点；再次，如果要让蚂蚁找到最短的路径，那么需要计算所有可能的路径并且比较它们的大小。事实上，每只蚂蚁并不是像我们想象的需要知道整个世界的信息，它们其实只关心很小范围内的眼前信息，而且根据这些局部信息利用几条简单的规则进行决策。

A. 基于蚁群算法的迷宫路径规划算法描述

(1)范围

蚂蚁观察到的范围是一个方格世界，蚂蚁有一个嗅觉范围 EyeShot（本程序中默认是 2），那么它能观察到的范围就是 EyeShot*EyeShot 个方格世界。

(2)环境

蚂蚁所在的环境是一个虚拟的世界，其中有障碍物，有别的蚂蚁，还有信息素，信息素有两种，一种是找到食物的蚂蚁洒下的食物信息素，一种是找到窝的蚂蚁洒下的窝的信息素。每个蚂蚁都仅仅能感知它范围内的环境信息。环境以一定的速率让信息素消失。

(3)觅食规则

在每只蚂蚁能感知的范围内寻找是否有食物，如果有就直接过去。否则看是否有信息素，并且比较在能感知的范围内哪一点的信息素最多，这样，它就朝信息素多的地方走，并且每只蚂蚁多会以小概率犯错误，从而并不是往信息素最多的点移动。蚂蚁找窝的规则和上面一样，只不过它对窝的信息素做出反应，而对食物信息素没反应。

(4) 移动规则

每只蚂蚁都朝向信息素最多的方向移，并且，当周围没有信息素指引的时候，蚂蚁会按照自己原来运动的方向惯性的运动下去，并且，在运动的方向有一个随机的小的扰动。为了防止蚂蚁原地转圈，它会记住最近刚走过了哪些点，如果发现要走的下一点已经在最近走过了，它就会尽量避开。

(5) 避障规则

如果蚂蚁要移动的方向有障碍物挡住，它会随机的选择另一个方向，并且有信息素指引的话，它会按照觅食的规则行为。

(6)播撒信息素规则

每只蚂蚁在刚找到食物或者窝的时候撒发的信息素最多，并随着它走远的距离，播撒的信息素越来越少。

B. 参数及其设置

本文主要用到的蚁群算法参数有：

- 最大信息素：蚂蚁在一开始拥有的信息素总量，越大表示程序在较长一段时间能够存在信息素，本文设为 10000。
- 信息素释放率：蚂蚁每走过一个点时，留下的信息素的数量占其拥有的信息素的比例，本文设置为 0.005。
- 信息素消失率：随着时间的流逝，已经存在于世界上的信息素会消减，这个数值越大，那么消减的越快，本文设置为 0.00002。
- 出错概率：蚂蚁不走信息素最大的区域的概率，越大则表示蚂蚁越有创新性，本文设为 0.2。
- 嗅觉范围：表示这个蚂蚁的感知范围，本为设置为 2。
- 路径记录最大值：表示蚂蚁能记住多少个刚刚走过点的坐标，这个值避免了蚂蚁在本地打转，停滞不前。而这个值越大那么整个系统运行速度就慢，越小则蚂蚁越容易原地转圈，本为设为 50。

IV. 迷宫路径规划算法的实现

为了保证所设计的迷宫路径是至少应保证有一条路径存在，因此，作者首先用深度遍历的方法初始化一条路径，然后再对剩余的非路径的点 用随机的方法设置可通或者不可通这样就能保证该迷宫最少有一条路径存在。

迷宫产生函数类主要包括：

- CreateMaze()//初始化迷宫函数
- SetNeighbor(int r, int c)；//初始化相邻参数
- SetAcross(int r, int c)；//用深度优先方法初始化一条通路
- CreateMore(int row ,int column)//随机初始化其他非路径节点
- DrawMaze(bool[,] H_line, bool[,] S_line)//绘画迷宫
- DrawPath(bool[,] Path)；//绘画各种路径

A. 基于栈的迷宫路径规划算法

传统的用栈的思想完成迷宫问题的思路:

(1)从入口开始，沿着随机的方向判断下一个节点是否可通过，如果是则将改点加入栈中。

(2)以原来的方向判断下一点是否可通过，如果是则继续仿照上一步的操作，如果不是则判断改点的其他方向上是否可通过如果是则该为其他方向。

(3)按照第一步和第二步的方法依次判断迷宫里的点直到找到一条最短路径结束。

基于栈的迷宫路径规划算法的主要功能函数有：

- SearchPath(bool[,] H_line, bool[,] S_line, int width)；//按照深度优先算法，借助于 栈完成路径搜索；
- GetPath(Point start, Point end)；//记录所得路径；
- Search(Point p)；//搜索某一点；
- MoveRole()；//根据规则，上下左右移动老鼠；

具体程序运行结果如图 2。

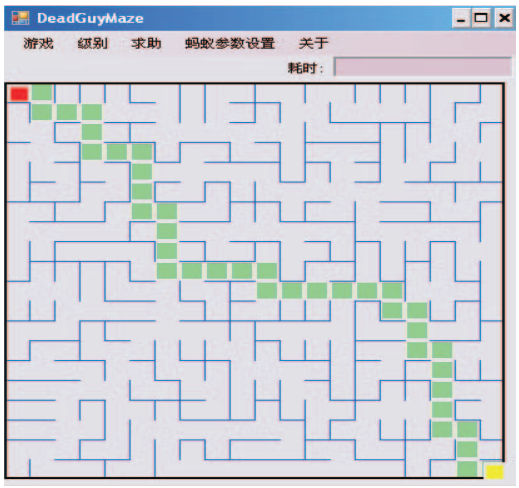


图 2　基于栈的迷宫路径规划算法运行结果

B.　基于蚁群算法的迷宫路径规划算法的实现

事先不告诉蚂蚁食物的位置，由左上角开始出发开始寻找食物，当一只找到食物以后，它会向环境释放一种信息素，吸引其他的蚂蚁过来，这样越来越多的蚂蚁会找到食物。有些蚂蚁并没有象其它蚂蚁一样总重复同样的路，他们会另辟蹊径，如果开辟的道路比原来的其他道路更短，那么，渐渐地更多的蚂蚁被吸引到这条较短的路上来。最后，经过一段时间运行，可能会出现一条最短的路径被大多数蚂蚁重复着。

基于蚁群算法的迷宫路径规划算法的基本函数包括：

- DropHomeSmell()；//释放家的信息素的方法
- DropFoodSmell()；//释放食物信息素的方法
- AntOneStep()；//蚂蚁移动的方法
- RandomDirection(int xxx, int yyy, AntDirection ddir)；/.获得蚂蚁前进方向地方法
- GetMaxSmell(int type, int xxx, int yyy, AntDirection ddir)；//获取信息素的方法
- JudgeCanGo(int xxx, int yyy)；//判断是否可行

运行结果如图 3 所示。

图 3 基于蚁群算法迷宫路径规划算法运行结果

V.　性能比较

本文主要从时间复杂度和运行结果进行分析。

A.　时间复杂度比较

基于栈的迷宫路径规划算法的时间复杂度是：

*T（n）=O(n*m*2)；*（n 和 m 分别道标迷宫的横向和竖向放个数）

而基于蚁群算法迷宫路径规划算法的时间复杂度是：

*T（n）=O（Nc*n2*m）*（n 和 m 分别道标迷宫的横向和竖向放个数）

B.　运行结果比较

当于迷宫障碍较多时即真正存在的路径较少时，设定运行时间为 10s。两种方法的解决能力有很大不同，基于栈的迷宫路径规划算法已经运行完毕如图 2 所示，而基于蚁群算法迷宫路径规划算法还没有找到，如图 3 所示。

图 4 基于蚁群算法迷宫路径规划算法运行 10s

但是在当迷宫障碍较少或者说路径很多的情况下采用一群算法却能很快的找到最短路径（效果如图 4 所示）而此时若采用最短路径算法却达不到蚁群算法所能达到的结果。

C.　运行结果分析

蚂蚁走迷宫的过程是一个群体性的活动因此若果存在较多的障碍则蚂蚁就会被分割道许多小空间中，因此它们之间信息的交流就很难起到作用，所以就算一只蚂蚁找到

了食物也因为信息素难以传递的缘故，无法让其他更多的蚂蚁参与进来。因而找最短路径的能力也下降了，但是当障碍较少时蚂蚁群体合作的效果就会显示出来因此他们找迷宫的速度会大大提高。

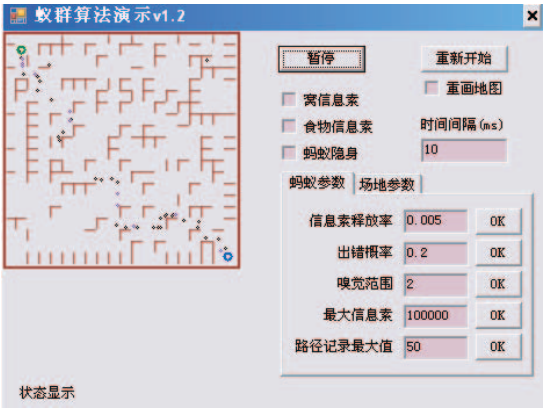


图 5 当障碍较少时求解迷宫最短路径的结果

VI. 结论

本文通过基于栈的迷宫路径规划算法和基于蚁群算法的迷宫路径规划的实验结果分析，得出结论：对于相同难度系数的迷宫若存在较多的路径即障碍较少时应该采用蚁群算法求解这样会得到更好的效果，而当迷宫中障碍较多时应该采用前者会得到较好的结果。因此，作者将在算法的通用性上进一步需求新的方法来解决。

References

[1] Ming Wang, Qian Zhang, The constrction of E-business: a case study. Journal of Peiking University, Vol 243, pp.102-103, April 2003 (In Chinese).

[2] Wen Ruchun,Xu Ying,Wang Zulin,Study and Application on Maze Path Planning Based on Improved Ant Colony Algorithm.Journal of Jianxi University of Science and Technology,Vol31, pp.26-30,Febrary 2010 (In Chinese).

[3] Hu Xiaobing,Huang Xiyue,Application of Ant Colony Algorithm to Maze Problem. Computer Simulation,Vol22,April,pp.58-60,2005(In Chinese).

[4] F.Ercal,H.C.Lee,Journal of Parallel and Distributed Computing Time-efficient Maze Routing Algorithms on Reconfigurable Mesh Architecutures,Vol.44,pp.133-140,1997

[5] S.Srinivasan,D.P.Mital,S.Haque,A Novel Solution for Maze Traversal Problems Using Artificail Neural Networks.Computer and Electrical Engineering,Vol.30,pp.563-572,2004

[6] Chen Yong-gang,Li Min,Fan Qing-Hui,Particle Swarm Optimization Algorithm to Solve Maze Problem.Journal of Henan University of Science and Technology:Natural Science,Vol 31, pp.51-53,85,Apr. 2010(In Chinese).

[7] Huang Meng,ec,Solution for Labyrinth Based on Rough Set and Genetic Algorithm.Modern Electronics Technique,Vol.24,pp.68-70,March,2009(In Chinese).

[8] Zhang Linfeng,Lv Hui,Qu Junfeng,New Algorithm for Solving Shortest Path of Maze Problem,Computer Engineer and Application,Vol.32,pp.263-265,Apral,2006(In Chinese).

[9] Liao Guo-yong,Wang Guang-chao,Labyrinth Problem s Solution with Genetic Algorithm, Journal of East China Jiaotong University, Vol.23,No.2,pp.138-140,Apr., 2006(In Chinese).