

学校代码	10406
分 类 号	TP391.9
密 级	
学 号	080081203101

盲审编号	238
------	-----

题目 基于蚁群算法的路径规划

研究

作者 华路

学 科、专 业 计算机应用技术

指 导 教 师 周之平

申请学位日期 2011 年 6 月

分类号: TP391.9

学校代码: 10406
学号: 080081203101

南昌航空大学
硕 士 学 位 论 文
(学位研究生)

基于蚁群算法的路径规划研究

硕士研究生: 华路

导 师: 周之平

申请学位级别: 硕 士

学科、专业: 计算机应用技术

所在单位: 信息工程学院

答辩日期: 2011 年 6 月

授予学位单位: 南昌航空大学

Based on ant colony algorithm for path planning research

A Dissertation

Submitted for the Degree of Master

On the Computer Applied Technology

By Hua Lu

Under the Supervision of

Prof. Zhou Zhiping

School of Information Engineering

Nanchang Hangkong University, Nanchang, China

June, 2011

摘 要

路径规划技术已经被广泛应用于飞行器、水面舰艇、地面车辆以及机器人等导航系统。目前求解路径规划问题的主要方法有 A*算法、遗传算法、人工势能场、神经网络、计算几何方法等。

蚁群算法是模拟蚂蚁觅食过程的一种仿生方法，将蚁群算法用于求解路径规划问题，近年来引起了国内外研究学者的广泛关注，也取得了一定的研究成果。

基本蚁群算法用于机器人路径规划容易出现早熟收敛，工作环境中障碍物分布密集时有可能规划不出合理的路径，在稀疏的连续环境下容易出现迂回搜索从而得不到最优路径。针对上述缺点，本文在基本蚁群算法的基础上提出了如下改进策略，以提高算法性能：(1)结合双蚁群算法和最大最小蚂蚁算法思想，对距离启发因子进行改进，增强目标点对蚂蚁的引导作用，避免迂回搜索；利用进化代数动态调整启发式系数 α ， β 和信息素挥发系数 ρ ，避免由于信息素的决定性作用使得进化后期出现早熟收敛；(2)利用终点距离信息初始化环境信息素以避免进化早期的盲目搜索，对不可行路径上的信息素进行分段线性调整并结合路径点回退策略以避免再次产生不可行路径，对状态转移概率排序并利用轮盘赌概率性选择路径点以提高算法的全局寻优能力。

利用 Visual C++ 6.0 开发二维平面路径规划实验仿真平台，在该平台上对本文算法与基本蚁群算法和其他改进蚁群算法进行仿真测试和性能比较。仿真结果表明，与同类算法相比，本文算法在运算时间和解的质量上都有较大程度的改善，表明了改进策略的有效性和可行性。

关键词：路径规划，蚁群算法，双蚁群算法

Abstract

Path planning technology has been widely applied in the aircraft, the surface vessels, ground vehicles and robot navigation system. The main methods which solving the path planning problem include AStar algorithm, genetic algorithm, artificial potential fields, neural networks, computational geometry method and so on.

Ant colony algorithm(ACO) is a biomimetic evolution method ,which mocks the process when ants seeking food, How to design ant colony algorithm for solving the robot path planning problem efficiently, has attracted much attention at home and abroad, meanwhile a lot of lectures emerges recently. As is known,when the basic ant colony algorithm is utilized to solve the robot path planning, it suffers from a large number of deficiencies, such as premature convergence, prone to bypass search under environment consisting of spacial sparsely continuous obstacles, and difficulty in providing a reasonable path for environments of high-density obstacles and so on. To overcome above shortcoming, some measures for enhancing performance are proposed on the basic of ant colony algorithm in this thesis as follows: (1) Double ant colony system is combined with Max-Min ant system and adjusting strategy for heuristic coefficient α , β according to the generations adaptively is introduced ,which drives ants move towards target directly for avoiding cycle searching and prevents from premature convergence efficiently.(2) Blind search in the early phases is inhibited by initializing the environmental pheromone with regard to the distance to the destination;infeasible paths are escaped from generating again by adjusting pheromone where as well as a retroversion strategy for moving away an illegal waypoint;the next waypoints selected from neighborhoods with a roulette based on the sorted ascendingly state transition probability in order to improve global optimization performance.

A path planning simulation platform in two-dimensional space is developed using visual C++ 6.0 environment, under which three algorithms such as standard ACO, other revised version and the new algorithms presented in this article are implemented and tested, the performance are compared each other sequentially. Simulation results show newly proposed algorithm in the dissertation performs over other counterparts in terms of between computation time and path quantity, as indicate that our improved strategies efficient and feasible.

Keyword: path planning, ant colony algorithm, double ant colony algorithm

目 录

摘 要	I
Abstract	II
目 录	IV
第一章 绪论	1
1.1 本课题研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 主要的环境建模方法.....	2
1.2.2 主要的路径规划算法.....	3
1.3 本论文的研究的主要内容.....	7
1.4 论文内容安排.....	8
1.5 本章小结	8
第二章 基本蚁群算法.....	10
2.1 基本蚁群算法的原理.....	10
2.1.1 蚁群行为描述.....	10
2.1.2 蚁群算法的基本原理.....	10
2.2 基本蚁群算法的数学模型.....	12
2.3 基本蚁群算法的性能评价.....	13
2.4 改进的蚁群算法.....	14
2.4.1 带精英策略的蚁群算法.....	14
2.4.2 蚁群系统(Ant Colony System).....	15
2.4.3 最大—最小蚂蚁系统(Max-Min Ant System, MMAS).....	16
2.5 蚁群算法的应用.....	17
2.6 本章小结	18
第三章 基于改进双蚁群算法的机器人路径规划.....	19
3.1 路径规划问题的环境表示.....	19
3.2 问题描述	20
3.2.1 环境描述.....	20
3.2.2 路径规划问题.....	22
3.3 基本蚁群算法用于路径规划的原理.....	22
3.3.1 节点选择规则.....	23
3.3.2 信息素更新规则.....	23
3.3.3 单蚁群算法的算法步骤.....	24
3.4 改进策略	24
3.4.1 改进启发式因子.....	25
3.4.2 信息素更新.....	25
3.4.3 权值系数 α 和 β 动态调整.....	26
3.4.4 挥发系数动态调整.....	26
3.5 双蚁群算法的步骤.....	27
3.6 仿真结果分析.....	28
3.6.1 仿真实例和参数设定.....	28
3.6.2 仿真结果与性能比较.....	28

3.7 本章小结	31
第四章 复杂环境下基于改进的蚁群算法机器人路径规划	32
4.1 基本蚁群算法的原理.....	32
4.1.1 节点选择规则.....	32
4.1.2 信息素更新规则.....	33
4.2 蚁群算法改进策略.....	33
4.2.1 距离启发因子的改进.....	33
4.2.2 基于排序的最优性选择下一个路径栅格.....	34
4.2.3 信息素初始化的改进.....	34
4.2.4 不可行路径信息更新规则.....	35
4.3 蚂蚁回退策略.....	36
4.4 改进算法步骤.....	37
4.5 仿真结果分析.....	38
4.5.1 仿真实例和参数设定.....	38
4.5.2 仿真结果与性能比较.....	38
4.6 本章小结	44
第 5 章 总结与展望	45
5.1 总结	45
5.2 展望	45
参考文献	47
攻读硕士学位期间论文发表和参加科研的项目情况.....	51
致谢	52

第一章 绪论

1.1 本课题研究背景及意义

随着计算机、电子和信息技术的发展，现代移动机器人和无人飞行器发生了巨大的变化，种类繁多，实现技术也越来越复杂。随着任务难度和危险度的不断上升，单纯靠手工操作实现移动机器人或无人飞行器的导航任务越来越困难。因此如何最大限度赋予移动机器人和无人飞行器智能，实现自主导航与控制，深受国内外研究人员的关注。路径规划作为移动机器人和无人飞行器自主导航的关键技术之一，已经成为目前国内外研究的一大热点。本课题主要针对平面环境中的移动机器人路径规划问题进行研究。

路径规划技术已经被广泛应用于飞行器、水面舰艇、地面车辆以及机器人等导航系统。机器人路径规划问题是机器人学的一个重要研究领域，它要求机器人依据一定的优化原则（如能量消耗最小、行走路线最短、行走时间最快等等），在其运动空间中找到一条从运动起点到目标点最优路径，且此路径能安全躲避障碍物。机器人已经被广泛的应用于教育、农业、军事、工业等社会的各方面，路径规划作为机器人系统的一个重要研究内容，其效果的好坏直接影响到机器人完成任务的质量好坏，所以很多国内外学者致力于研究开发高效的、新颖的机器人路径规划方法。移动机器人路径规划也是机器人应用中的一项重要技术，例如，在执行抢险、装配、焊接、探测等任务时，采用完善的移动机器人路径规划技术可以大量节省机器人作业时间、减少机器人的磨损和减少人工探测的危险，与此同时，可以节约人力资源和降低成本，为机器人在较多种行业中的应用奠定良好的基础。通过智能型路径规划方法，可以提高机器人路径规划的避障精度，加快路径规划速度，满足各种实际应用的需要。我国在智能移动机器人研究方面虽然已经取得了部分成果，如地面飞行机器人、水下自主机器人和自主导航车等，但由于我国在智能移动机器人方面的研究起步较晚，在研究和应用方面都落后于一些西方发达国家，而且有些方面还没有达到完全实用。因此，进行移动机器人路径规划算法的研究，具有一定的理论价值和工程应用意义。

1.2 国内外研究现状

路径规划是指移动目标在有障碍物的工作环境中，在给定起点和目标点的条件下，寻找一条最短的运动路径，并且要求移动目标沿该路径运动能安全地避开障碍物的威胁。目前路径规划问题可归结为两类：一类是环境信息完全已知的全局路径规划，另一类为局部环境信息可知的局部路径规划。全局路径规划又可以分为环境建模和搜索策略两部分。环境建模的主要方法有栅格法、自由空间法、拓扑表示法、可视图法等方法。根据处理方法的不同，路径规划算法主要分为几何算法、启发式算法、基于物理过程的算法和智能算法等算法。

1.2.1 主要的环境建模方法

（1）自由空间法

自由空间法^[1,2]采用预先定义的基本形状构造自由空间，基本形状如广义锥形和凸多边形，并将自由空间表示为连通图，通过连通图的搜索来进行路径规划。自由空间的构造：从障碍物的一个顶点开始，依次作到其它的顶点的连接线，删除不必要的连接线，使得连接线与障碍物边界所围成的每个自由空间都是面积最大的凸多边形。连接各连接线的中点形成的网络图就是机器人可自由移动的路线。此方法的优点比较灵活，起始点和目标点的改变不会造成连通图的重构，但其缺点是该算法的复杂度与障碍物的多少成正比，并不是任何情况下都能得到最优路径。

（2）可视图法

可视图法^[1,2]忽略了机器人的大小，将机器人作为点处理，将机器人的起始点、目标点和多边形障碍物的所有顶点进行组合相连，要求机器人的起始点和障碍物各顶点之间的连线，目标点和障碍物的各顶点之间的连线，各障碍物顶点与顶点之间的连线，均不能穿越工作空间中的障碍物，即直线被称为“可视”的。通过此方法进行环境建模，路径寻优问题就转化为从机器人起始点到目标点经过这些“可视”直线的最短距离问题。通过一些优化算法可以删除一些可视图图中不必要的连线以此简化可视图，减少机器人路径寻优的搜索时间。该方法适合用顶点表示的多边形障碍物，对圆形障碍物该方法无效。当起始点和目标点发生改变，需要重新构造可视图，比较麻烦。李善寿等人^[3]提出基于改进可视图法的环境建模，该方法与传统的可视图方法不同，不是把障碍物的顶点作为图的节点，而是把障碍物顶点连线的中点作为节点，把这些节点间的某些边线作为弧，最后通过遗传

算法进行全局规划。

(3) 栅格法

栅格法^[4, 5, 6]是将机器人的工作环境分解成一系列具有二值信息的网格单元，一般多采用四叉树或八叉树方式来表示机器人的工作环境。工作环境中障碍物的位置和大小一致，且机器人运动过程中，障碍物的位置和大小不发生变化。用相同大小的栅格对机器人的二维工作环境空间进行划分，栅格大小以机器人自身的大小为准。某个栅格范围内不包含障碍物，此栅格为自由栅格，反之为障碍栅格。该方法是以栅格为单位记录环境信息，环境被量化成具有一定分辨率的栅格，栅格大小直接影响信息存储量的大小和路径规划时间的长短。栅格划分较大，环境信息存储量就较小，路径规划时间较短，且在大规模复杂的环境下搜索路径的能力减弱；栅格划分较小，环境分辨率较高，在大规模复杂的环境下搜索路径的能力较强，但环境信息存储量较大，路径规划时间较长。栅格粒度大小的确定是栅格法面临的主要问题，袁增任等人^[7]将改进栅格法与回归预测两种方法相结合，使得该算法适用于静态和动态障碍物环境中的移动机器人路径规划问题。文献[7]对栅格的改进采用以障碍物为单位，记录的信息量大大减少，克服了栅格法中环境信息存储量大的问题。

(4) 拓扑法

拓扑法将工作空间划分成具有拓扑特征的子空间，根据子空间彼此的连通性建立拓扑网络，在其网路上寻找从起始点到目标点的拓扑路径，最终由拓扑路径得出几何路径。拓扑法的主要思想是降维，将在高维几何空间中求路径的问题转化为在低维拓扑空间中辨别连通性的问题。其方法的优点是利用拓扑特征很大程度上缩小了搜索空间，算法的复杂度完全依赖于障碍物的数目多少，该方法在理论上是完备的，而且拓扑法通常不需要明确机器人的准确位置，对于位置误差就有了更好的鲁棒性；该方法的缺点是在拓扑网络的建立过程非常复杂，尤其是当障碍物增加时，如何更有效地修正已经存在的拓扑网络以及如何提高图形速度是有待解决的问题。

1.2.2 主要的路径规划算法

(1) 几何算法 基于可视图的机器人路径规划忽略机器人的大小，将机器人视为一质点，在起始点和障碍物的顶点、目标点和障碍物的顶点以及障碍物顶点之间构造连接线，使得连接线均不能穿越障碍物，即直线称为是“可视”的。路径寻优问题就是寻找出从起始点到目标点经过这些“可视”直线的最短距离问题。

杨淮清等^[8]针对自由空间法存在的不足,提出可视图法用于机器人路径规划。该方法通过将复杂轮廓的障碍物近似地看作矩形或多个矩形的组合,以此建立描述障碍物的边界图,再根据建立的边界图实现机器人路径规划。该方法克服了自由空间法对环境要求苛刻的缺点,增强了对环境的适应性。其缺点是将复杂障碍物近似看作矩形,得到的路径不一定是最优的路径。

基于 Voronoi 图的机器人路径规划中, Voronoi 图由两个或多个障碍给定特征元素距离相等的点集构成。连接机器人的起点和目标点以及已经构造好的 Voronoi 图,构成环境中自由空间的框架图。在此基础上,通过利用搜索 Voronoi 图的边界确定起始点和终点的最短路径。与可视图法相比,构造 Voronoi 图所需要的时间比可视图要少的多。Voronoi 图更倾向于使图中机器人与障碍物距离最大化,并且路径较光滑平顺,但不能保证路径最优。

基于三角剖分的机器人路径规划算法中通过将机器人所在的工作空间中的自由空间进行三角剖分,此路径规划算法就是通过图搜索算法寻找出起点到终点的最短路径,在环境复杂的情况下三角剖分法找不到最短的路径。

(2) 启发式算法

A*算法中的启发函数是由起始点到当前点的最小目标函数与当前点到目标点的估计目标函数值计算得到的,A*算法通过对开放链表中启发函数值最小的节点依次扩展,形成一个具有父子关系的路径点集,这些排列有序的路径点集构成了一条从起点到终点的路径。

A*搜索算法是一种经典的启发式搜索算法^[9]。Hasan 等人^[10]利用地形数据库构造了数字地图,并采用了一种修正的 A*算法来搜索最佳的位置。该算法的代价函数由距离、危险和机动能力约束的加权和形式表示,并且上述代价函数中各元素的大小用模糊隶属度函数表示,使规划的路径、航迹上的障碍以及所需机动的内在不确定性得到较好的反映。为了加快搜索过程,Szczerba 等人^[11,12]提出一种用于二维规划的稀疏 A*算法。该算法有效减少搜索空间,实时收敛且得到满足要求的解,提高收敛速度,节省了内存空间。国内学者李春华等^[13,14],进一步将稀疏 A*算法扩展到三维规划应用中。Stentz^[15,16]针对自主式机器人在只有部分已知或完全未知的环境中的最优路径规划问题,提出了一种动态 A*算法—D*算法。D*算法通过维护一个搜索树,局部地更新代价图以重新扩展相关节点来获得环境发生变化时的最优路径,可保证规划的最优性和实时性。

(3) 基于物理过程模拟的算法

基于物理过程模拟的算法主要有人工势场法和模拟退火算法。

人工势场法最早由 Khatib 于 20 世纪 80 年代中期提出的^[17],其思想是将移动

的机器人视为带电粒子在一种虚拟力场中运动，目标点对移动机器人产生引力，障碍物对移动机器人产生斥力，移动机器人根据目标点和障碍物点对机器人产生的合力来确定自己运动路径。该方法的优点是规划速度快，适合实时规划。缺点有可能找不到路径，其原因是在吸引力和排斥力相等的地方存在局部最小点，出现障碍物时会产生振荡，导致不稳定。

针对复杂环境下的机器人路径规划问题，庄晓东等人^[18]提出了一种基于人工势场和蚁群算法的混合路径规划算法，该方法利用人工势场规划路径结果作为先验知识，对蚂蚁的路径进行初始化，以提高蚁群算法的性能。该算法不但克服了人工势场法容易产生的局部最优问题，而且也提高了蚁群算法路径规划的效率。其缺点是通过人工势场法规划的路径结果作为先验知识计算量较大。石为人等人^[19]通过改进传统人工势场中的斥力函数，确保目标点是机器人势场为全局最小。为了克服势场法容易陷入局部极小，提出了一种连接局部最小区域障碍物的方法，建立机器人离散传感器模型。张建英等人^[20]通过改进势场函数确保目标点的势场是全局最小值，引入附加控制力避免陷入局部极小。之后。张建英等^[21]在此基础上进行改进，通过选择适当的增益系数克服了目标不可达问题，并通过拉紧法去除路径冗余节点，得到一条极坐标下对称多项式圆滑的最优路径。

模拟退火算法^[22]来源于固体退火原理，将固体加温到一定的温度，固体内部分子随着温度的升高变为自由活动的无序状态，使其内能增大，随着温度的缓慢下降，分子运动趋于有序，在每个温度都达到平衡态，最后在常温时达到基态，内能减为最小。根据 Metropolis 准则，分子在温度 T 时趋于平衡的概率为 $\exp(\Delta E/KT)$ ，其中 E 为温度 T 时的内能， ΔE 为其改变量， K 为 Boltzmann 常数。王仲民等人^[22]提出一种将模拟退火算法和共轭方向法相结合的新型混合优化算法，该算法可以避免陷入局部最优而得到最优解。其优点是计算简单，收敛速度快。

(4) 智能算法

智能算法主要有遗传算法、粒子群优化算法和蚁群算法。

遗传算法是由美国学者 Holland 于 20 世纪 70 年代中期首先提出来的^[23]，它是一种基于自然群体遗传演化机制的高效探索算法。该算法通过模拟自然界的生物进化过程，采用人工进化的方式对目标空间进行随机搜索。它将问题中的可行解看作群体中的个体或染色体，并将染色体编码，通过对群体反复进行遗传、交叉、变异，通过目标适应度函数对个体进行评价，依据适者生存，优胜劣汰的进化规则不断得到更优的群体。该方法运算速度较慢，需要占据较大的存储空间和运算时间。

李庆中等人^[24]提出基于遗传算法的移动机器人动态避障路径规划算法，通过

将复杂的二维路径编码问题简化成一维编码问题，并把路边约束和动态避障要求和最短路径要求融合在一个简单的适应度函数。其算法优点：可以实时的实现动态避障，能为机器人提供一条较好的局部路径。王强等人^[25]提出基于遗传算法的移动机器人的一种路径规划方法，通过两阶段的遗传操作进行动态避障路径规划。其优点实时性较强，且能得到较优的路径。

周兰凤等^[26]提出了一种基于知识的遗传算法用于机器人路径规划。该算法将知识领域并入初始种群，对遗传参数进行自适应调整，较好地克服了早熟收敛问题，提高了算法的效率。

Kennedy 和 Eberhart 等^[27]于 20 世纪 90 年代提出粒子群优化算法，该算法模拟鸟群飞行觅食的行为，通过鸟群之间的集体协作使群体到达目的。在该算法中，每个备选解称为“粒子”，多个粒子共存、合作寻优，每个粒子根据它自身的“经验”和相邻粒子群的最佳“经验”在问题空间中向更好的位置“飞行”，搜索最优解。该算法的优点具有收敛速度快，算法简单，容易编程实现等特点，但该算法也有缺点，如容易陷于局部极值点，导致得不到全局最优解。

秦元庆等人^[28]提出基于粒子群算法的多阶段路径规划方法。该算法首先通过连接图建立机器人工作的环境模型，然后利用 Dijkstra 算法求得连接图最短路径，再采用粒子群算法对 Dijkstra 算法求得最短路径进行优化。其优点能满足移动机器人导航的高实时性要求。谭冠政等人^[29]提出基于粒子群算法的移动机器人全局最优路径规划，该方法通过栅格法对环境建模，直接采用粒子群算法搜索全局最优解。该算法收敛速度快，但不能保证全局最优。孙波等^[30]提出基于粒子群优化算法的移动机器人全局路径规划。该算法首先对机器人工作环境建模，然后通过坐标变换建立新地图，再利用粒子群算法规划一条最优路径。算法运行效率高、收敛速度快，但不能保证得到的路径为最优路径。

蚁群优化算法 (Ant Colony Optimization: ACO) 是 Coloni 和 Dorigo 等^[31-34]在 20 世纪 90 年代初提出的一种新型分布式智能仿生类算法，它模拟和借鉴了现实世界中蚂蚁种群的觅食行为特征。蚂蚁在运动时会在通过的路径上释放一种特殊的分泌物——信息素来寻找路径。当它们碰到一个还没有走过的路口时，就随机地挑选一条路径前行，同时释放出与路径有关的信息素。蚂蚁走的路径越长，则释放的信息量越小。当后来的蚂蚁再次碰到这个路口的时候，选择信息量较大的路径概率相对较大，这样便形成了一个正反馈机制。最优路径上的信息量越来越大，而其他路径上的信息量却会随时间的流逝消减，最终整个蚁群会找出最优路径。该方法具有可并行性、寻优能力强、适应性强和易于其他算法结合等优点，但也存在一定缺陷，如搜索时间长、对复杂问题的描述能力不强和容易出现搜索

停滞等等。

Dorigo 于 1991 年提出了蚂蚁系统^[35, 36], 1996 年提出了蚂蚁系统算法^[37], 1999 年提出了蚁群算法^[38, 39], 同年 T.Stitzle 等人提出了最大最小蚂蚁系统^[40, 41]。Walter 和 Thomas 先后对蚁群算法的收敛性进行了证明^[42-45]。国内学者对蚁群算法的研究也取得了部分成果。丁建立等^[46]将遗传算法全局收敛性和蚂蚁算法的正反馈特性的相结合, 用于机器人路径规划。吴斌等^[47]提出了基于蚂蚁算法的 TSP 问题分段求解算法。攀晓平等^[48]提出了复杂环境下基于蚁群算法的机器人路径规划方法, 通过提供路径点选择的多种可选策略和权值系数的动态调整对传统蚁群算法进行改进, 以规划出一条更为合理的路径。朱庆保等^[49]提出一种基于蚁群算法的机器人动态路径规划算法。该算法采用栅格法建立环境模型, 通过蚁群算法规划出一条初步路径, 结合基于滚动窗口的环境探测和碰撞检测规划局部避碰路径。

1.3 本论文的研究的主要内容

目前, 对路径规划的问题的研究已取得了较多的成果, 主要通过一次性的全局规划得到一条从起点到终点的安全轨迹, 并采用某些算法对路径进行优化, 如模拟退火、禁忌搜索、遗传算法、神经网络和粒子群算法等, 但这些算法存在搜索空间大、算法复杂、效率不高等问题。采用蚁群算法实现路径规划, 为路径规划提供了新的解决思路。由于传统的蚁群算法存在收敛速度慢, 容易陷入局部最优等缺点。本课题提出改进的蚁群算法用于机器人路径规划。研究的重点在于如何改进算法, 可以提高算法的收敛速度和得到更优的解。

本课题是基于传统蚁群算法的改进用于移动机器人的路径规划中, 研究的主要内容如下:

(1) 传统蚁群算法中能见度启发因子 η_{ab} 只有两种取值 1 或 $\sqrt{2}$, 能见度启发因子对蚂蚁寻找路径启发性并不明显, 通过改进传统蚁群算法能见度启发因子, 增强距离的影响力度, 从而提高蚁群算法的收敛速度, 避免陷入局部最优; 在传统蚁群算法中 α 为信息激素物质的相对重要性; β 为能见度的相对重要性, 传统蚁群算法这两个参数是固定不变的, 由于传统算法蚂蚁在进化早期下一个路径点的信息素量和距离启发因子两方面差别都很小, 通过动态设置参数, 增大可选路径点的状态转移概率差别避免盲目搜索; 传统蚁群算法中挥发系数是为定值, 通过动态改变挥发系数, 避免挥发因子太大导致随机搜索, 太小时陷入局部最优。

(2) 传统蚁群算法对不可行的路径信息素更新采用传统的局部更新, 这种方式的更新极有可能导致蚂蚁陷入“死胡同”, 得到的解为局部最优, 算法容易死锁。

通过动态更新不可行路径信息素，对其不可行和可行路径信息素采用不同的更新策略，引入回退策略，避免再次产生不可行路径，防止算法死锁；传统蚁群算法中蚂蚁对下个可行节点的选择一般采用的是随机选择方式，通过排序的轮盘赌最优选择方法，避免迂回，使蚂蚁对下个节点的选择更有确定性；传统的蚁群算法在路径规划中将路径各节点的信息素初始为一固定常数，使蚂蚁开始阶段盲目选择下一个路径节点。通过终点距离信息初始化环境信息素，避免算法进化早期的盲目搜索。

1.4 论文内容安排

本文共分为五个部分，其内容安排如下：

第一章 绪论

在参考较多文献和资料的基础上首先介绍了本课题的研究背景和意义，其次介绍了关于本课题的国内外研究现状，最后介绍了本文的主要研究内容。

第二章 基本蚁群算法的原理

首先介绍基本蚁群算法的原理和数学模型，其次介绍了蚁群算法的性能评价，然后介绍了各种蚁群算法如带精英策略的蚁群算法，基于排序的蚁群算法，最大最小蚂蚁算法，最后介绍了蚁群算法的各种应用领域。

第三章 基于改进的双蚁群算法用于机器人路径规划

首先介绍路径规划中环境建模问题，通过对机器人的工作环境栅格化的方式进行环境建模，其次本文在基本蚁群算法的基础上提出了四种改进策略，最后本文通过提出改进的双蚁群算法与基本蚁群算法和其他双蚁群算法的实验结果比较，验证改进的双蚁群算法的可行性和有效性。

第四章 基于改进的蚁群算法用于机器人路径规划。

首先在基本蚁群算法的基础提出了四种改进策略，防止算法过早的收敛，防止出现死锁。其次通过实验仿真比较改进后的蚁群算法与基本蚁群算法和第三章提出的改进的双蚁群算法，验证此种改进算法的有效性。

第五章 总结和展望

对全文的工作进行了总结，并在此基础上提出一些尚需要完善和深入研究的问题，并展望此算法在以后路径规划的应用前景。

1.5 本章小结

本章首先介绍了本课题的研究背景及意义，其次介绍了本课题的国内外研究

现状，然后介绍了本论文的主要研究内容，最后介绍了本论文的内容安排。

第二章 基本蚁群算法

基本蚁群算法是由 20 世纪 90 年代意大利学者 Marco Dorigo 等人^[31]首次提出来的算法,即蚂蚁系统(Ant System)。蚁群算法是继遗传算法、人工势场和模拟退火等算法的又一种新颖的启发进化算法。它的应用领域越来越广泛,不仅用于离散问题的优化,且也可用于连续问题的优化。在本章中,主要介绍基本的蚁群算法的原理及其改进算法。

2.1 基本蚁群算法的原理

2.1.1 蚁群行为描述

尽管单个蚂蚁个体比较简单,但蚁群却表现出高度结构化的社会组织,蚁群在较多情况下能完成单个蚂蚁所不能完成的复杂的任务,这种处理复杂任务的能力来源于蚂蚁群体的个体之间的相互协作。

以蚂蚁寻找食物为例说明蚂蚁如何寻找路径觅食。在自然界中,蚂蚁的食物源总是随机散落在蚁巢周围。由于单个蚂蚁的能力和智力很简单,不可能单独找到食物源。但它们通过相互协作能找到食物源到巢穴的之间的最短路径。蚂蚁在寻找食物时通过释放在路径上一种特殊的分泌物——信息素(pheromone)来寻找路径。当蚂蚁经过一个还未走过的路口时,以随机的方式挑选一条路径前行,同时释放出与路径有关的信息素。蚂蚁行走的路径越长,则释放的信息量就越小,反之,释放的信息量就越大。当后来的蚂蚁再次经过这个路口的时候,选择信息量较大的路径概率相对较大,这样便形成了一个正反馈机制。最优路径上的信息量越来越大,随着时间的流逝其他路径上的信息量却会减少,最终整个蚁群会沿着信息素浓度最高的路径行走,最终找到巢穴离食物源最短的路径。

2.1.2 蚁群算法的基本原理

蚁群算法觅食路径搜索原理如图 2-1 所示。

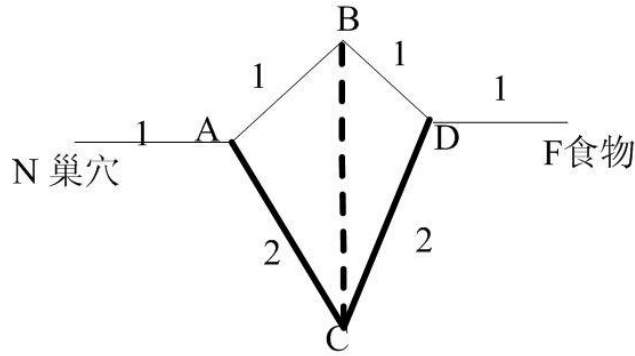


图 2-1 蚂蚁算法基本原理流程图

图 2-1 中，假设蚂蚁从巢穴到食物源存在两条可行路径，一条为 N-A-B-D-F，长度为 4；另一条为 N-A-C-D-F，长度为 6。其中 AB，BD 长度为 1，AC 和 CD 长度为 2，NA 和 DF 的长度为 1，蚂蚁匀速运动，速度为 1 个单位长度，蚂蚁寻找食物前路径上没有信息素。蚂蚁寻路过程如下：

$t=0$ 时刻，一组 20 只蚂蚁从巢穴出发移动到 A，由于所有路径上信息素浓度为 0，则它们以相同的概率选择路径 ABD 和 ACD 的，所以 20 只蚂蚁中 10 只走路径 ABD，另 10 只走路径 ACD。

$t=4$ 时刻，走路径 ABD 的蚂蚁到了食物源，将开始折回，而此时走路径 ACD 的蚂蚁到达 CD 的中点。

$t=5$ 时刻，两组蚂蚁在 D 点相遇。由于 BD 上的信息素浓度和 CD 上信息浓度相同，导致各有 10 只蚂蚁选择了相应的路径。折回的 10 只蚂蚁，5 只选择了 BD，另 5 只选择了 CD，走路径 ACD 的 10 只蚂蚁继续向食物方向移动。

$t=8$ 时刻时，前 5 只蚂蚁返回巢穴，此时在 B 点，AC 的中点和 CD 的中点各有 5 只蚂蚁。

$t=9$ 时刻，前 5 只蚂蚁又回到 A 并且要重新选择走路径 ABD 还是路径 ACD。此时 AB 上的轨迹数为 10，而 AC 上的轨迹数为 15，由于信息素的驱使，较多的蚂蚁选择路径 ABD，从而使路径 ABD 的信息素更高。随着时间的推移，两条路径上的信息素浓度的差距越来越大，直至最后绝大多数蚂蚁选择了最短路径 ABD。以上就是蚂蚁觅食能找到最短路径的原理。

受到自然界中真实蚁群集体行为的启发，意大利学者 M.Dorigo 于 1991 年首次提出了基于蚂蚁种群的新型优化算法—蚁群算法，并用该方法解决了一些组合优化问题。在蚁群算法中提出了人工蚂蚁的概念。人工蚂蚁具有两重特性，一是真实蚂蚁行为特征的抽象，将蚁群觅食行为最关键的部分赋予了人工蚂蚁，二是人

工蚂蚁有真实蚂蚁不具备的本领。人工蚂蚁与真实蚂蚁一样，是一群相互合作的个体，通过相互合作得到较优的结果。人工蚂蚁与真实蚂蚁一样有共同的任务就是寻找最短路径。人工蚂蚁与真实蚂蚁一样使用信息素进行间接通讯。

2.2 基本蚁群算法的数学模型

以求解旅行商(TSP)问题为例，说明基本蚁群算法的模型。旅行商问题是一个NP问题，利用传统的穷举法很难求出问题的最优解。蚁群算法作为一种智能算法被用于求解TSP问题，且能得到较为满意的解。

旅行商问题就是指给定 n 个城市和两两城市之间的距离，要求确定一条最短路线，此路线当且仅当经过每个城市一次。其图论描述为：给定图 $G=(V,A)$ ，其中 V 为顶点集， A 为各顶点相连组成的边集，且已知各顶点的连接距离，要求确定一条长度最短的 Hamilton 回路，即遍历所有顶点当且仅当一次最短回路。

为了模拟真实蚂蚁的行为，首先引入如下标记：

m 为蚁群中蚂蚁的数量；

d_{ij} 为城市 i 和 j 之间的距离；

$b_i(t)$ 为 t 时刻位于城市 i 的蚂蚁个数， $m = \sum_{i=1}^n b_i(t)$ ；

η_{ij} 为边 (i,j) 的能见度，反映城市 i 转移动 j 的启发程度，这个量在蚂蚁系统的运行中不变；

τ_{ij} 为边 (i,j) 上的信息素浓度；

$\Delta\tau_{ij}^k$ 为蚂蚁 K 在边 (i,j) 上留下的信息素量；

p_{ij}^k 为蚂蚁 K 的转移概率， j 是尚未访问的城市；

α 为信息素启发式因子；

β 为期望启发式因子；

ρ 为信息数挥发系数。

基本蚁群算法步骤：

Step1:初始化，设置相关参数，初始时刻各条路径上的信息素量相等。设 $\tau_{ij}(0)=C$ ， $\Delta\tau_{ij}(0)=0, N_C=0$ ， N_C 为循环次数，最大循环次数为 N_{\max} ；

Step2:循环次数 $N_C=N_C+1$ ；

Step3:将蚂蚁随机分布在 N 个城市上；

Step4:蚂蚁从当前城市 i 选择下一个要访问的城市 j 时，根据概率公式(2-1)进行选择，每只蚂蚁将访问的城市存放到各自的禁忌表 tabu_k 中，如此循环直至所有的蚂蚁访问完所有城市；如公式(2-1)所示。

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha(t)\eta_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t)\eta_{is}^\beta(t)} & s \in allowed_k \\ 0 & s \notin allowed_k \end{cases} \quad (2-1)$$

其中 $allowed_k = \{C-tabu_k\}$ 表示蚂蚁 K 下一步允许选择的的城市集合； $\eta_{ij}(t) = 1/d_{ij}$ 。

Step5: 计算每只蚂蚁行走的总路径长度，保存最优解。

Step6: 根据式 (2-2)、(2-3) 进行信息素更新。

$$\tau(j) = (1 - \rho)\tau_{ij} + \Delta\tau_{ij} \quad (2-2)$$

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (2-3)$$

式中 $\Delta\tau_{ij}^k$ 表示第 K 只蚂蚁在本次循环中留在 (i,j) 上的信息量。

Step7: 如果 $N_c < N_{max}$ ，则清空禁忌表转 **step2**；否则循环程序结束并输出程序计算结果。

根据信息素更新方式的不同，M.Dorigo 提出了三种不同的基本蚁群算法模型分别称之为蚁周(Ant-cycle)模型、蚁密(Ant-Density)模型和蚁量(Ant-Quantity)模型。

在蚁密系统模型中：

$$\Delta\tau_{ij}^k = \begin{cases} Q & \text{若第} K \text{只蚂蚁在本次循环中经过路径}(i, j) \\ 0 & \text{否则} \end{cases} \quad (2-4)$$

蚁量系统模型

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{d_{ij}} & \text{若第} K \text{只蚂蚁在本次循环中经过}(i, j) \\ 0 & \text{否则} \end{cases} \quad (2-5)$$

d_{ij} 为城市 i 和城市 j 之间的距离；

蚁周系统模型

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{若第} K \text{只蚂蚁在本次循环中经过}(i, j) \\ 0 & \text{否则} \end{cases} \quad (2-6)$$

其中 Q 表示信息素强度，为一大于零的常数。 L_k 表示第 k 只蚂蚁在本次循环中所走路径的总长度。

2.3 基本蚁群算法的性能评价

为了全面衡量基本蚁群算法的优劣程度，引入了三个基本指标评价基本蚁群算法的性能^[52]。

(1) 最佳性能指标

定义相对误差 E_0 为最佳性能指标，其公式如 (2-7) 所示。

$$E_0 = \frac{c_b - c^*}{c^*} \times 100\% \quad (2-7)$$

式中, c_b 表示算法多次运行所得到的最佳优化值; c^* 表示所求问题的理论最优值, 当理论最优值未知时可用已知最佳优化值来代替。最佳性能指标用以衡量基本蚁群算法对问题的最佳优化度, 其值越小意味着蚁群算法的优化性能越好。

(2) 时间指标

定义基本蚁群算法的时间性能指标 E_t , 其公式如 (2-8) 所示。

$$E_t = \frac{I_a T_0}{I_{\max}} \times 100\% \quad (2-8)$$

式中, I_a 表示算法多次运行后, 满足终止条件时的迭代次数平均值; I_{\max} 表示给定的最大迭代次数; T_0 表示算法一次迭代的平均计算时间。时间性能指标用以衡量基本蚁群算法对问题解的搜索快慢程度, 在 I_{\max} 固定的前提下, E_t 越小说明蚁群算法的收敛速度越快。

(3) 鲁棒性能指标

定义基本蚁群算法的鲁棒性能指标 E_r , 公式如 (2-9) 所示。

$$E_r = \frac{c_a - c^*}{c^*} \times 100\% \quad (2-9)$$

式中, C_a 表示算法多次运行后所得到的平均值; c^* 表示所求问题的理论最优值。鲁棒性能指标用以衡量基本蚁群算法对随机初值和操作的依赖程度。其值越小说明算法的鲁棒性越强。

基本蚁群算法的综合性能指标 E 可以表示为上述三个指标性能的加权组合, 如公式 (2-10) 所示。

$$E = a \cdot E_0 + b \cdot E_t + c \cdot E_r \quad (2-10)$$

式中, a, b, c 分别表示最佳性能指标、时间性能指标和鲁棒性能指标的加权系数, 且满足 $a+b+c=1$;

E 值越小则说明算法的综合性能越好。

2.4 改进的蚁群算法

2.4.1 带精英策略的蚁群算法

带精英策略的蚁群算法^[51](Ant algorithm with elitist strategy)是最早改进的蚂蚁系统。精英策略类似于遗传算法中的精英保留。在遗传算法中, 通过基因的选择、交叉、变异产生下一个个体。在下一代个体中最优个体可能不被保留。蚁群算法

中引入精英策略，为了使目前为止所找到的最优解在下次循环中对蚂蚁更有吸引力，在每次循环以后给予最优解额外的信息素量。这样的解为全局最优解，找到这个解的蚂蚁称为精英蚂蚁。信息量更新如式(2-11)所示。

$$\begin{aligned}\tau_{ij}(t+1) &= \rho\tau_{ij}(t) + \Delta\tau_{ij} + \Delta\tau_{ij}^* \\ \Delta\tau_{ij} &= \sum_{k=1}^m \Delta\tau_{ij}^k\end{aligned}\quad (2-11)$$

其中

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_K} & \text{若第K只蚂蚁在本次循环中经过}(i, j) \\ 0 & \text{否则} \end{cases}\quad (2-12)$$

其中

$$\Delta\tau_{ij}^* = \begin{cases} \sigma \cdot \frac{Q}{L^*} & \text{如果边}(i,j)\text{是属于找出的最优解的一部分} \\ 0 & \text{否则} \end{cases}\quad (2-13)$$

其中 $\Delta\tau_{ij}^*$ 表示精英蚂蚁引起的路径 (i,j) 上的信息素量的增加； σ 是精英蚂蚁的个数； L^* 为所找出的最优解得路径长度。

使用精英策略的蚂蚁系统可以比传统的蚂蚁系统找到更好解，但如果精英蚂蚁数量过多，搜索会集中在极值周围从而导致算法早熟收敛。

2.4.2 蚁群系统 (Ant Colony System)

蚁群系统^[52]是由 Dorigo 和 Gambardella 在 1996 年提出的，用于改善蚁群算法的性能。

蚁群系统在基本的蚁群算法的基础上做了如下三个方面的改进：

(1)改进状态转移规则

在基本的蚁群算法中，蚂蚁完全依靠概率进行路径的选择。蚂蚁使用的是随机比例规则，有倾向性的寻找新的路径。在蚁群系统中，蚂蚁使用了伪随机比例规则，寻找新的路径。这种规则既可以利用问题的先验知识，即城市之间的距离启发信息和路径上信息素量。同时也可以有倾向性的探索新路径。伪随机比例规则如公式(2-14)所示。

$$j = \begin{cases} \arg \max_{u \in allowed_k} \{[\tau(i,u)]^a [\eta(i,u)]^\beta\} & q \leq q_0 \\ S & \text{否则按式(2-1)} \end{cases}\quad (2-14)$$

式中, q 是在 $[0,1]$ 区间之间均匀分布的随机数， q_0 是一个事先设定的常数 ($0 \leq q_0 \leq$

1) , S 为式(2-1)给出的概率。 q_0 的大小决定了利用先验知识与寻找新路径之间的相对重要性：每当一只蚂蚁在城市 i 在选择下一个城市 j 时，它选择一个随机数 q 。如果 $q \leq q_0$, 则根据先验知识选择最好的边，否则按式(2-1)选择其它一条边。

(2)全局信息素更新改进

在蚁群算法中全局更新时更新所有蚂蚁所走的路径，这样降低了蚂蚁搜索最优路径的效率，使蚂蚁较难以较快的速度在最优路径附近集中搜索。而在蚁群系统中，在每次循环中只更新最优蚂蚁所走过的路径，最优蚂蚁所走过的路径上的信息得到较大的增强，其他路径上的信息素随着时间的消逝逐渐在挥发，这就增大了最优蚂蚁上的路径和其他较差路径上的信息上的差异，使得蚂蚁更倾向于选择最优路径，从而使其搜索很快集中在最优路径附近，提高了算法的搜索效率。全局信息素更新如式(2-15)所示。

$$\begin{aligned} \tau(i, j) &= (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \Delta\tau(i, j) \\ \Delta\tau(r, s) &= \begin{cases} (L_{gb})^{-1} & \text{如果 } (i, j) \text{ 属于全局最优路径} \\ 0 & \text{否则} \end{cases} \end{aligned} \quad (2-15)$$

其中 α 为全局信息素挥发系数。

(3)局部信息更新

在蚁群算法中只进行全局信息更新，并没有进行局部信息更新。而蚁群系统每次构造解得同时进行局部信息更新，每次循环以后，同样进行全局信息素更新。局部信息更新如式(2-16)所示。

$$\tau(i, j) = (1 - \rho) \cdot \tau(r, s) + \rho \cdot \Delta\tau(i, j) \quad (2-16)$$

其中， ρ 为一个局部信息素挥发系数， $0 < \rho < 1$ 。

2.4.3 最大—最小蚂蚁系统(Max-Min Ant System, MMAS)

最大最小蚂蚁系统在基本蚁群算法基础上主要做了如下三个方面的改进：

(1) 基本蚁群算法信息更新是对所有蚂蚁走过的路径进行信息素更新，而最大最小蚂蚁系统是对当前循环最优的蚂蚁或者全局最优的蚂蚁进行信息素更新，其他蚂蚁走过的路径只是进行信息素挥发。在 MMAS 中只有一只蚂蚁用于每次循环后的信息更新。更新规则如式 (2-17) 所示。

$$\begin{cases} \tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}^{best} \\ \Delta\tau_{ij}^{best} = 1/f(s^{best}) \end{cases} \quad (2-17)$$

其中 $f(s^{best})$ 表示当前迭代最优解或者全局最优解。在 MMAS 中主要使用迭代最优解，如果使用全局最优解，会使算法集中在全局最优解附近搜索，减少了蚂

蚁搜索范围，容易使算法陷入局部最优。使用迭代最优解每次循环得出的值并不相同，使更多解的元素获得信息素增强，扩大了蚂蚁搜索范围，得出的结果更优。

(2) 信息素限制

MMAS 中信息素更新无论采用迭代最优还是全局最优解，都可能导致算法搜索的停滞。如果某些节点的信息过高，明显高于其他节点，导致节点上信息素之间的浓度差距较大，这有可能导致算法早熟收敛，算法有可能会停滞。MMAS 为了防止各节点之间信息素差异过大，对各个节点信息限制在 $[\tau_{\min}, \tau_{\max}]$ 区间内。

(3) 最大最小蚂蚁系统初始化时将所有路径上的信息素设置为 τ_{\max} ，而基本蚁群算法并没有这样的设置。

2.5 蚁群算法的应用

蚁群算法自问世以来已经得到广泛的应用，其中最成功的就是在组合优化问题中的应用。根据优化问题的性质可分为两类，一类应用是静态组合优化问题，如旅行商问题、车间调度问题、二次分配问题、车辆路线问题等；另一类应用于动态组合优化问题，如网络路由问题。

(1) 旅行商问题是 NP 问题之一，蚁群算法最初用于旅行商问题。旅行商问题就是给定 N 个城市，寻找一条通过 N 个城市一次且只有一次的回到起点的最短路径。较多研究表明，蚁群算法用于解决 TSP 问题优于模拟退火、遗传算法、神经网络等优化算法^[53]。

(2) 二次分配问题^[54]就是将 n 个设备分配给 n 个位置，使得分配的的代价最小化。代价就是将设备分配到位置上的方式函数。二次分配问题是一般化的 TSP 问题，所以可以将蚁群算法用于解决二次分配问题。

(3) 车间作业调度问题(job-shop scheduling problem,JSP)也是一个典型的 NP-hard 问题。JSP 问题是存在 J 个工作和 M 个机器，每个工作有一系列的操作组成，如何求得从第一个操作开始到最后一个操作结束之间的最小时间间隔，要求操作严格遵循串行顺序，且在特定的时间每个操作在特定的机器完成，在同一时刻每台机器不能完成不同的工作。蚁群算法继遗传算法、禁忌搜索算法、模拟退火算法又一个新的进化算法用于 JSP 问题优化^[55]。

(4) 路径规划问题就是运动物体在某种约束条件下，规划出最短路径问题。例如蚁群算法用于移动机器人路径规划问题^[56]，蚁群算法用于旅行商问题等^[57]。

(5) 车辆路线问题(VRP)VRP 问题是车辆数目和每辆车的容量已知，主要问题是找出最佳行车路线，且此路线在满足某些约束条件下使得运输成本最小。利用

蚁群算法优于模拟退火算法和神经网络算法。

(6)网络路由问题在某些约束条件动态寻找一条最短路径问题。惠普公司和英国电信公司开展了这方面的研究，提出了蚁群路由算法，每只蚂蚁根据它在网络上的经验和性能，动态更新路由表项。

2.6 本章小结

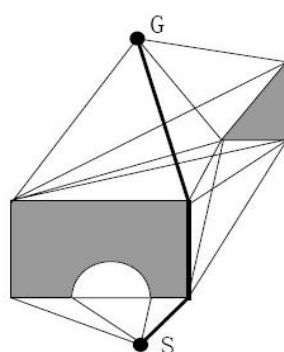
本章首先详细介绍了蚁群算法的基本原理，且以 TSP 问题为例说明了蚁群算法的数学模型，并以 TSP 问题介绍了基本蚁群算法的算法过程，其次介绍了蚁群算法的性能衡量标准，然后介绍了几种改进的蚁群算法，最后对蚁群算法的应用做了介绍。通过本章对基本蚁群算法的详细介绍，对下一章对基本蚁群算法的改进奠定了基础。

第三章 基于改进双蚁群算法的机器人路径规划

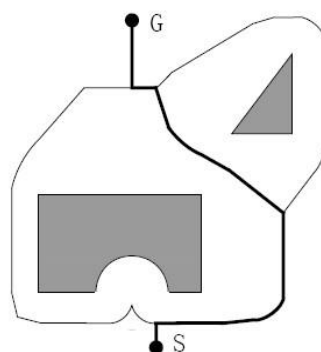
路径规划就是要求移动机器人在障碍物的工作环境中，寻找一条从起点到终点的路径，且这条路径既要使机器人能安全躲避障碍物，而且要在一定的优化目标下（如距离、时间、能量等）尽可能的优化。目前使用的方法主要 Petri 网算法、基于数据融合的模糊规划、神经网络算法、人工势场、计算几何法和遗传算法等，但以上算法在复杂的环境下进行路径规划时，存在明显的不足。比如算法的计算量较大，有时得不到较好的解。基本的蚁群算法用于路径规划搜索路径能力不强，容易早熟收敛，且很容易陷入局部最优。为此，提出了改进距离启发因子、信息素更新，权值系数动态调整和挥发系数线性调整等四种改进策略，以提高算法的性能。

3.1 路径规划问题的环境表示

机器人在路径规划前必须建立环境的表示方法，对机器人活动空间的有效描述称为环境模型。将机器人的工作环境空间由原始的形式通过转化为适合规划的模型，这个过程称为环境建模。环境建模主要是障碍物的表示。合理的环境模型可以减少机器人路径规划的搜索量，有利于减少存储空间。



(a) 可视图



(b) Voronoi 图

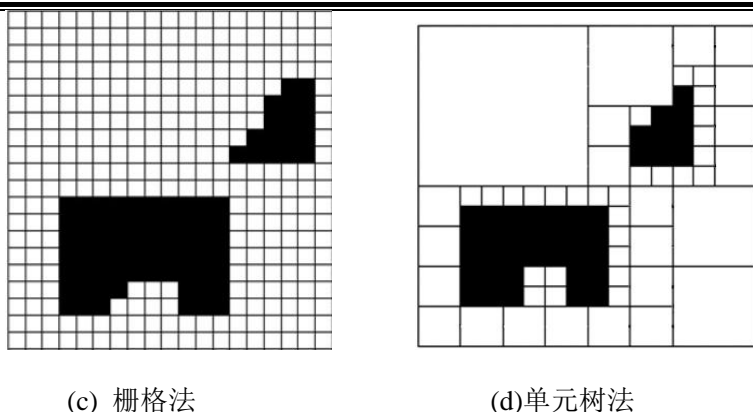


图 3-1 四种常用的环境建模方法

下面主要说明环境的表示方法。如图 3-1 中(a)图为用可视图的表示的环境，可视图的构造：将起始点、目标点和多边形障碍物的所有顶点进行组合相连，要求起始点和障碍物各顶点之间、目标点和障碍物的各顶点之间以及各障碍物顶点与顶点之间的连线，均不能穿越障碍物，即直线是“可视”的。其缺点是建模时间较长，较难用于实时规划。图 3-1 中(b)图为 Voronoi 图的环境表示，Voronoi 图是由两个或多个障碍的给定特征元素距离相等点的集合构成。Voronoi 图的构造时间比可视图耗时要少，此方法的缺点是规划的路径较长，机器人偏离障碍物较远。图 3-1 中(c)图为栅格法环境建模，栅格法建模以机器人的大小为栅格大小标准，用大小相同的栅格来划分机器人工作空间，一般用栅格数组来表示工作环境。图 3-1 中(c)图黑色栅格代表障碍物栅格，白色为自由栅格，自由栅格为机器人可行走的栅格。其优点是方法简单，适合大规模并行处理机制实现，其缺点表示效率不高，栅格的大小的设定会影响到机器人的路径规划效果。图 3-1 中(d)图为单元树法，该方法将机器人的工作空间划分为较大的单元，一般来说二维空间划分为四部分，称为四叉树；三维空间划分为八部分，称为八叉树。每个单元所占用的空间有三种情况，都是自由空间，都是障碍物空间和混合型空间。对于混合空间按照先前的方法继续划分，直到设定好的精度为止。其缺点是计算单元之间邻接关系时的损失。

3.2 问题描述

3.2.1 环境描述

由于采用栅格法建模比较直观且建模相对较容易，因此本文采用栅格法建模。栅格法建模的思想：设机器人的活动工作空间为矩形，内部分布多个静止的障碍

障碍物为矩形，以机器人长、宽的最大值作为机器人的步长，将活动空间划分为等间距的栅格图。设栅格的大小为单位长度，含有障碍物的栅格为障碍栅格，不含障碍物的栅格为自由栅格，自由栅格为机器人可行栅格。栅格的大小决定了栅格划分的地图中记录的路径信息与实际路径信息的误差。采用栅格法建模使机器人的连续的路径被离散化，机器人的运动轨迹被分解为离散的单个运动，单个运动被记录在每个栅格。考虑建模的复杂性，一般实际运用上规定移动机器人可以有四个方向（上、下、左、右）的移动，如图 3-2 所示。或者八个方向的移动（上、右上、右、右下、下、左下、左、左上），如图 3-3 所示。

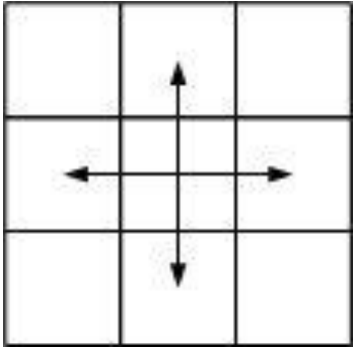


图 3-2 机器人四种移动方向

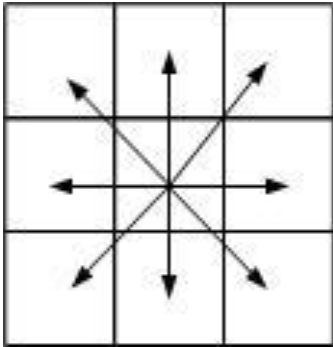


图 3-3 机器人八种移动方向

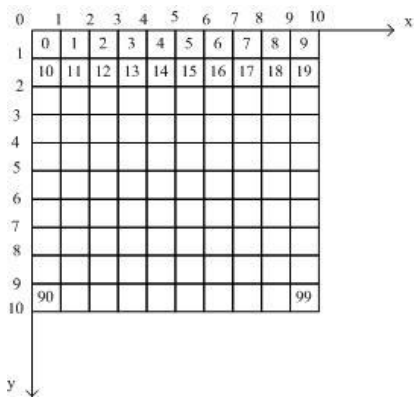


图 3-4 栅格坐标与序号的关系

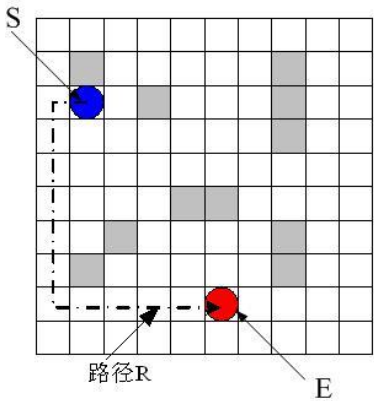


图 3-5 合理路径图

对划分的栅格采用序号法进行编号，如图 3-4 所示。

设 AS 为机器人 rob 在二维平面上的有限运动区域，假设机器人运动区域分布有限个障碍物。在 AS 中建立直角坐标系，且以 AS 左上角为最表点 O，横向为 x 轴，纵向为 y 轴，x,y 方向上的最大值分别为 X_{max}, Y_{max} 。设机器人的步长为 Ra，以步长为单位长度对 x,y 进行划分，每行的栅格 $N_x = X_{max}/Ra$ ；每列的栅格数 $N_y = Y_{max}/Ra$ 。如果障碍为不规则的形状，进行栅格划分，当不满一个栅格的时，

算一个栅格。

栅格序号值与其坐标的对应关系如式(3-1)所示。

$$\begin{cases} x = f(x, y) \bmod N_x \\ y = f(x, y) / N_y \\ f(x, y) = y \times N_x + x \\ g(i, j) = g(x+1/2, y+1/2) \end{cases} \quad (3-1)$$

其中 $f(x,y)$ 为横坐标为 x , 纵坐标为 y 的栅格的序号值; N_x 为每行的栅格数; N_y 为每列的栅格数, \bmod 为求余运算。 $g(i,j)$ 为栅格中心点坐标, 地图中的自由栅格用 0 表示, 障碍物栅格用 1 表示。机器人路径规划就是从起点到目标点的路径规划就转化成从一个自由栅格节点到另一个自由栅格的节点的移动。

3.2.2 路径规划问题

路径规划问题是指在给定有障碍物的环境空间中, 搜索从起点到终点的无障碍最短路径。在实际过程中, 为了便于计算, 通常采用栅格化模型简化表示环境空间, 即将环境空间分割为大小相等的栅格, 以栅格的中心作为路径点规划出一条合理路线, 如图 3-5 所示。在图 3-5 中灰色栅格表示障碍物, 白色栅格表示为自由栅格, S 所指栅格为起点栅格, E 所指栅格为终点栅格。

规划问题的数学表示形式如式(3-2)所示。

$$\begin{aligned} \min \quad & \sum_{g_i, g_{i+1} \notin \mathbf{O}} d(g_i, g_{i+1}) \\ s.t \quad & \\ & g_i \vec{g}_{i+1} \cap \mathbf{O} = \Phi, \forall i \end{aligned} \quad (3-2)$$

其中 $d(g_i, g_{i+1})$ 表示两相邻路径点 g_i, g_{i+1} 的欧氏距离, \mathbf{O} 为障碍物占用的栅格集。

3.3 基本蚁群算法用于路径规划的原理

蚂蚁在觅食过程中在它所经过的路径上留下一一种信息素物质, 且在运动过程中对其信息素进行感知, 以此指导自己的运动方向。由大量蚂蚁组成的蚁群集体表现为一种信息的正反馈现象, 路径越短, 该路径上走过的蚂蚁越多, 则留下的

信息素浓度越大，后来者选择该路径的概率相对就越大，蚂蚁个体之间在通过这种信息的交流来选择最短路径并达到搜索食物的目的，蚁群算法就是模拟蚁群在这一行为的优化算法。

3.3.1 节点选择规则

蚂蚁 K 在运动过程中，会根据每条路径上的信息素浓度决定下一步的转移方向，在 t 时刻，如图 3-6，设蚂蚁 K 在点 $A(i,j)$ 处，待选节点 $B(k,l)$ ， t 时刻蚂蚁由位置 A 转移到 B 的概率如式(3-3)所示。

$$p_{ab}^k(t) = \begin{cases} \frac{\tau_{ab}^\alpha(t) \eta_{ab}^\beta(t)}{\sum_{s \in M_{ab(i,j)}^k} \tau_{as}^\alpha(t) \eta_{as}^\beta(t)} & s \in M_{ab(i,j)}^k \\ 0 & s \notin M_{ab(i,j)}^k \end{cases} \quad (3-3)$$

式中 η_{ab} 为线段 AB 的能见度，为点 A 和点 B 之间距离的倒数，即 $1/d_{ij}$ ； τ_{ab} 为线段 AB 上的信息素浓度， α 为信息激素物质的相对重要性； β 为能见度的相对重要性， $M_{ab(i,j)}^k$ 为当前栅格节点 A 下一步待选栅格节点的集合。

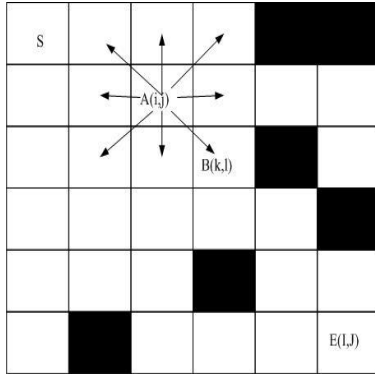


图3-6环境栅格模型

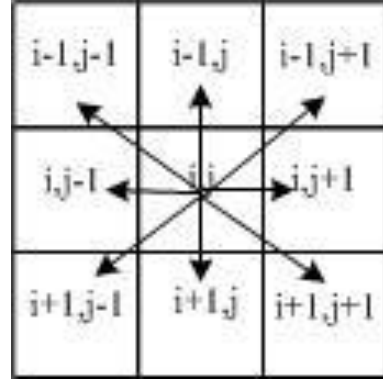


图3-7 可行栅格示意图

为防止蚂蚁在栅格化环境中走斜线穿越障碍物的顶点，还需要增加如下约束条件^[60]：如图3-7，若点 $(i-1, j)$ 为障碍物点，则点 $(i-1, j-1)$ 、 $(i-1, j+1)$ 为不可行栅格；若点 $(i, j+1)$ 为障碍物栅格，则点 $(i+1, j+1)$ 、 $(i-1, j+1)$ 为不可行栅格；若点 $(i+1, j)$ 为障碍点，则点 $(i+1, j+1)$ 、 $(i+1, j-1)$ 为不可行栅格点；若点 $(i, j-1)$ 为障碍物点，则点 $(i-1, j-1)$ 、 $(i+1, j-1)$ 为不可行栅格。

3.3.2 信息素更新规则

在一次循环中，当所有蚂蚁都到达目的点，应用信息素更新规则，只对走过

的节点进行更新，其他节点的信息素只是进行信息素挥发。信息素更新规则如(3-4)所示。

$$\tau(j) \leftarrow (1 - \rho) \tau(j) + \Delta \tau(j) \quad (3-4)$$

在式(3-4)中 ρ 为信息素挥发因子 ($0 < \rho < 1$) ; $\Delta \tau(j)$ 为全局更新因子，如 (3-5) 所示。

$$\Delta \tau(j) = \sum_{k=1}^M \Delta \tau_k(j) \quad (3-5)$$

其中：

$$\Delta \tau^k(j) = \begin{cases} \frac{Q}{L_K} & \text{蚂蚁K经过节点j} \\ 0 & \text{蚂蚁K不经过节点j} \end{cases} \quad (3-6)$$

式(3-6)中， Q 为信息量常数， L_K 为蚂蚁 K 在本次循环中搜索到的路径长度。

3.3.3 单蚁群算法的算法步骤

- Step1:** 初始化参数，设置路线的起点和终点，为每一个栅格赋初始信息素 c ，形成信息素矩阵，设置最大循环数为 N_{\max} ；
- Step2:** 将 m 只蚂蚁放在起点，将起点存放在蚂蚁各自的禁忌表 tabu_k 中；
- Step3:** 每只蚂蚁按照式(3-3)的概率 p_{kab} ，进行下一个节点的选择；
- Step4:** 循环执行 **Step3**，直到所有蚂蚁到达目标点；
- Step5:** 计算每只蚂蚁得到路径的长度，根据蚂蚁得到的路径长度，依据信息素调整准则(式(3-4~3-6))对各节点的信息素进行更新；
- Step6:** 循环执行步骤 **Step2** ~ **Step5** 直到得到路径最优解或者循环次数达到设定的最大次数停止。

3.4 改进策略

传统单蚁群算法用于机器人路径规划容易早熟收敛，陷入局部最优，搜索范围较小，搜索性能较低。在复杂的凹环境和大规模障碍物环境下传统的单蚁群算法容易陷入“死胡同”，导致算法停滞，规划不出较合理的路径。在稀疏的障碍物环境中，传统蚁群算法容易陷入局部搜索，走不出障碍物区间，也会导致算法停滞，得不出较合理的路径。基于此本文提出利用双蚁群算法，通过双蚁群算法的路径交叉，对其中一种群的路径进行优化，规划出较优路径，避免算法在复杂环

境下出现停滞。为此，在传统蚁群算法的基础上引入最大最小蚂蚁系统，并从启发式因子、信息素挥发系数和信息素更新方式等方面进行改进。具体细节如下文所述。

3.4.1 改进启发式因子

蚁群算法是一种启发式算法，启发因子的好坏对算法起较大的作用。在传统的单蚁群算法中，所使用的距离启发因子是栅格节点*i*和栅格节点*j*之间的距离的倒数，即为 $1/d_{i,j}$ ， $d_{i,j}$ 为节点*i*到节点*j*之间的距离。单蚁群使用 $1/d_{i,j}$ 来区分 $M_{ab(i,j)}^k$ 内要选择的节点，采用这种方式对各栅格节点区分不明显，造成蚂蚁盲目搜索。因为相邻栅格的距离不是栅格的边长，就是栅格边长的 $\sqrt{2}$ 倍。当点(*i,j*)与目标点相距较远时，距离启发因子所起的作用甚微，实际效果并不是很理想。文献[48]提出的改变距离启发因子，只考虑了当前节点四个方向的邻域，其他四个方向的邻域并没考虑到，规划的结果有可能得到局部最优解，同时文献[48]还引入了三个标定系数，标定系数的选取对结果也有影响。文献[58]提出的改变距离启发因子，存在计算时间长，空间耗费大的缺点。该文在此基础上提出改进的距离启发因子，如图 3-6 中定义 $L_{(m,n)}$ 为点 $B(m,n)$ 到目标点 $E(I,J)$ 距离， $L_{m,n} = \sqrt{(I-m)^2 + (J-n)^2}$ ，启发因子为 $\eta_{ab}=1/L_{(m,n)}$ 。

改进后的概率选择公式如式 (3-7) 所示。

$$p_{ab}^k(t) = \begin{cases} \frac{\tau_{ab}^\alpha(t)\eta_{ab}^\beta(t)}{\sum_{s \in M_{ab(i,j)}^k} \tau_{as}^\alpha(t)\eta_{as}^\beta(t)} & s \in M_{ab(i,j)}^k \\ 0 & s \notin M_{ab(i,j)}^k \end{cases} \quad (3-7)$$

$L_{(m,n)}$ 的值越小，说明离目标点越近，距离启发因子越大，B点被选择的概率相对较大。

3.4.2 信息素更新

传统的单蚁群算法所采用信息素更新方式，会出现有一些路径节点上信息素量过高，正反馈效果过强，使更多的蚂蚁集中在信息量过高的路径上，容易陷入局部最优，也会出现过早的收敛，并且搜索范围减小。同时也会出现一些路径节点信息素过少，甚至接近于0，从而使一些路径不在被搜索到，也可能失去最优解的路径。基于以上不足，该文结合最大最小蚂蚁系统的算法思想，进行路径信息更新。在最大最小蚁群系统中只有一只蚂蚁用于每次循环后信息素更新，此蚂蚁可能是当前循环中最优的蚂蚁或者是全局最优的蚂蚁，其更新规则如下^[59]：

$\tau_{ij}(t+1)=\rho\tau_{ij}(t)+\Delta\tau_{ij}^{best}$, 其中, $\Delta\tau_{ij}^{best}=Q/f(s^{best})$, $f(s^{best})$ 表示周游蚂蚁或全局最优蚂蚁走的路径值。选择周游最优的蚂蚁还是全局最优的蚂蚁来进行信息素更新, 虽然可以加快算法规划路径的速度和提高路径规划的质量, 但算法只更新周游最优的蚂蚁或全局最优的蚂蚁, 使算法较快集中较优解周围, 算法很快收敛, 这样将导致蚂蚁搜索范围减小, 得到的解有可能是局部最优解, 所以通过对信息素轨迹的最小值和最大值分别施加了 τ_{min} 和 τ_{max} 限制, 从而使得对所有信息素轨迹 $\tau_{min} \leq \tau_{ij} \leq \tau_{max}$, 避免蚂蚁陷入局部最优。利用此思想对每次循环后每只可行解蚂蚁所走的路径进行信息素更新, 采用如下方式更新信息素, 如果每个节点路径上信息素更新后为 τ_{ij}^{new} , $\tau_{ij}^{new}=(1-\rho)\tau_{ij}^{old}+\rho\Delta\tau$, 如果 $\tau_{ij}^{new} < \tau_{min}$ 则 $\tau_{ij}^{new} = \tau_{min}$, 否则如果 $\tau_{ij}^{new} > \tau_{max}$, 则 $\tau_{ij}^{new} = \tau_{max}$ 。

3.4.3 权值系数 α 和 β 动态调整

参数 α 和 β 分别决定信息素轨迹和距离启发函数的相对重要性。在传统单蚁群算法中, α 和 β 为常数, 在算法执行过程保持不变。在问题规模较小的路径规划中, α 和 β 参数固定, 对结果的影响并不明显。但是在稍微大规模的路径规划中, α 和 β 参数固定不变, 对结果有所影响。传统单蚁群算法在初始化时, 每个路径上初始为相同的信息素, 在最初的循环中信息素启发系数对蚁群的启发性并不明显, 不能明显区别下一步要走的节点, 使蚂蚁盲目搜索路, 而最初的循环中起主要作用的是距离启发系数, 它对下个节点的选取有明显的启发性, 更容易选择节点到目标节点比较短的节点。本文提出 α 和 β 随循环次数变化而调整, 如公式(3-8)和公式(3-9)所示。

$$\alpha = \begin{cases} 3i/100 & 0 < i \leq n \\ 2.5 & n < i \leq N_c \end{cases} \quad (3-8)$$

$$\beta = \begin{cases} 3-1.5i/100 & 0 < i \leq n \\ 2 & n < i \leq N_c \end{cases} \quad (3-9)$$

式(3-8)和式(3-9)中 i 为循环次数, n 为循环次数的临界点, N_c 为最大的循环次数。在刚开始的 n 次循环中, 距离启发系数起主导作用, 信息素启发系数启发作用不明显。临界点 n 的选择有多次实验确定, 本文设定 n 次循环内, 随着循环次数的递增, α 为线性递增, β 为线性递减的, 在 n 次循环以后, α 和 β 均为常数。通过这种方式的改进, 可以避免算法陷入局部最优, 得到路径更优。

3.4.4 挥发系数动态调整

信息素挥发系数对蚂蚁路径全局搜索有一定的影响。由于信息素挥发系数 ρ

的存在, 使未被搜索到的节点上信息素减少到 0, 降低算法的全局搜索能力, ρ 过小时, 以前搜索过的节点被再次选择的可能性比较大, 也会影响全局搜索能力。 ρ 过大会降低算法的收敛速度。挥发系数 ρ 保持不变, 使蚂蚁搜索能力降低, 有部分节点很难搜索到, 得到的解未必是全局最优解。本文提出动态更新信息素挥发系数, 挥发系数动态调整的公式如 (3-10) 所示。

$$\rho(t+1) = \begin{cases} \rho(t)a^{-\frac{n}{N_c}} & \rho(t) > \rho_{\min} \\ \rho_{\min} & \rho(t) \leq \rho_{\min} \end{cases} \quad (3-10)$$

其中 a 为常数, n 为循环次数, N_c 为最大的循环次数, ρ 为初始化时为常数, 当 ρ 降到 ρ_{\min} 时, 保持 ρ 恒定不变。在算法开始阶段信息挥发系数 ρ 较大, 随着循环次数的增大, 挥发系数逐渐的减小, 通过此种改进能扩大蚂蚁的搜索范围, 避免陷入局部最优。

以上改的改进策略将用双蚁群算法, 双蚁群算法的步骤如下。

3.5 双蚁群算法的步骤

- Step1:** 初始化参数, 设置路线的起点和终点, 为每一个栅格赋初始信息素 τ_{\max} , 设置最大循环数 N_{\max} 和步数极限 steplimit ;
- Step2:** 将种群 (1) 的 m 只蚂蚁放在起点上, 将种群 (2) 的 m 只蚂蚁放在目标点上, 对种群 (1) 和种群 (2) 中每只蚂蚁对下一个节点的选择都按照式 (3-3) 的概率 p_{kab} , 使用轮盘赌算法进行节点的选择, 并把得到的节点存放到各自的禁忌表中;
- Step3:** 循环执行 **Step2**, 直到种群 (1) 所有蚂蚁到达目标点, 种群 (2) 所有蚂蚁到达起点, 在规定的步数内没有到达各自目标的点蚂蚁标记为 “死亡”;
- Step4:** 通过路径交叉得到最优种群的路径^[58], 保存最优种群每只蚂蚁可行蚂蚁所行的路径长度;
- Step5:** 由最优种群可行蚂蚁所走的路径长度 L_K , 计算可行蚂蚁走过路径上各点分得的信息素为 Q/L_K ;
- Step6:** 对最优种群所有蚂蚁所走过的路径通过 (式 (3-4~3-6)) 进行信息素更新, 更新后的信息素为 $\tau_{(i,j)}^{\text{new}}$, 如果 $\tau_{(i,j)}^{\text{new}} < \tau_{\min}$ 则 $\tau_{(i,j)}^{\text{new}} = \tau_{\min}$, 如果 $\tau_{(i,j)}^{\text{new}} > \tau_{\max}$, 则 $\tau_{(i,j)}^{\text{new}} = \tau_{\max}$;
- Step7:** 循环执行步骤 **Step2 ~ Step6**, 直到得到最优解或者循环次数达到最大。

3.6 仿真结果分析

3.6.1 仿真实例和参数设定

实例 1：随机模型，栅格数为 30×30 ,障碍物比率 $p=0.05,0.1,0.3,0.5$;

实例 2：凹形障碍模型，栅格数分别为 15×15 、 30×30 ;

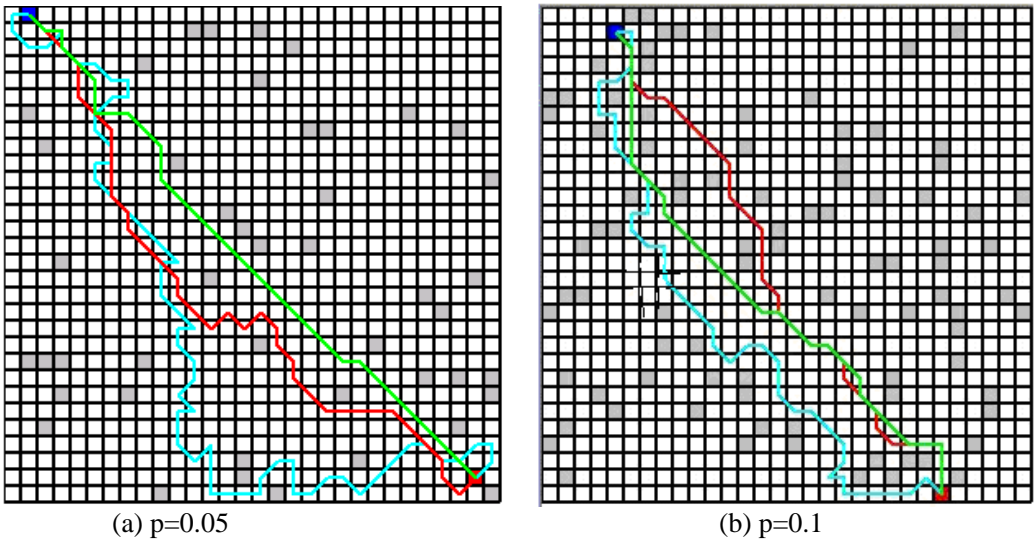
上述 4 种实例的起点、目标点和环境空间的配置分别如图 3-5 所示，栅格大小为 10，仿真参数如下表 3-1 所示。

表 3-1 仿真参数表

α	β	ρ	M	N_{\max}	steplimit	τ_{\min}	a	ρ_{\min}	Q	τ_{\max}	n
2.0	2.5	0.7	20	300	250	20	2	0.3	1000	400	100

3.6.2 仿真结果与性能比较

为了验证性能,对上述 4 种实例进行测试,在 Windows XP 系统下(CPU 2.0GHZ, 内存 1GB) VC++ 6.0 开发环境下，分析比较本文所提出改进的双蚁群算法与蚁群算法（ACA：Ant Colony Algorithm）、文献[58]提出的双蚁群算法的性能。为表述方便，将本文算法简称为 IDACA：(Improved Double Ant Colony Algorithm)，文献[58]提出的双蚁群算法简称为：DACA(Double Ant Colony Algorithm)，三种算法路径规划的路径如图(3-8~3-9)所示，图中蓝色表示蚁群算法（ACA），绿色表示改进的双蚁群算法(IDACA),红色为文献[58]提出的双蚁群算法(DACA)。



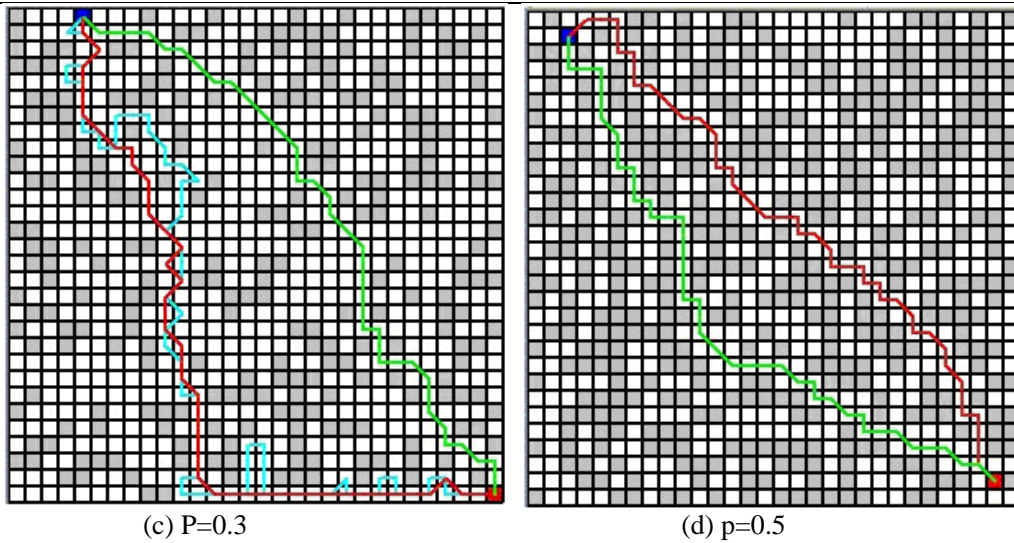
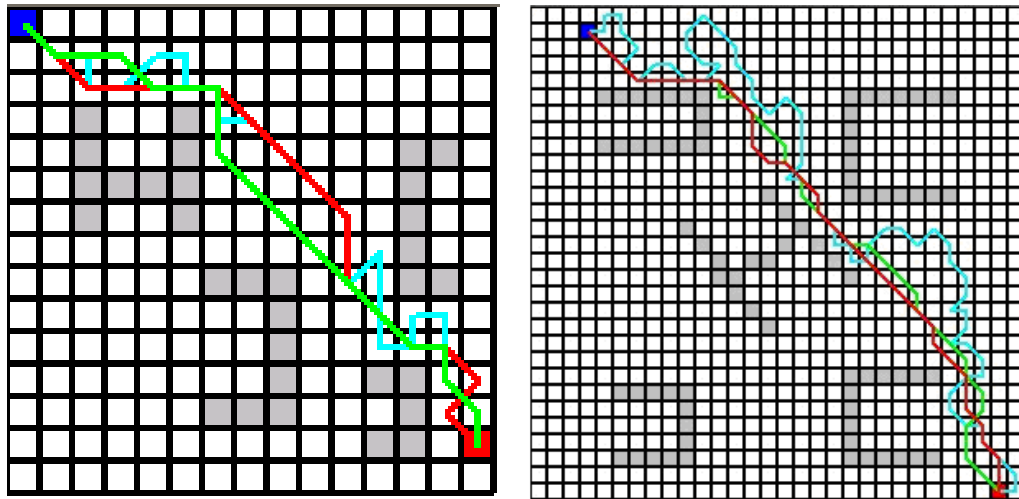
图 3-8 仿真区域为 30×30 栅格时，随机障碍物环境下的仿真结果比较（ p 为障碍物比例）

图 3-9 凹形障碍物仿真结果

从图 3-8 可以看出 ACA 规划出的路径最长，且路径上存在多处尖角，路线曲折。DACA 规划的路径好于 ACA，且与 IDACA 规划的路径有部分重合，但也有多处弯折。IDACA 规划出的路径更平滑，尖角较少，规划的出的路径更理想。在图 3-8 (d) ACA 规划不出较合理的路径，而 IDACA 可以规划出较优的路径。原因是：在复杂的环境下，可行的路径比较少，且环境中存在多处“死胡同”，ACA 在复杂的环境下容易导致蚂蚁陷入“死胡同”，规划不出合理的路径。在算法刚开始阶段由于部分蚂蚁走入“死胡同”，部分蚂蚁在所走过路径上留下信息素，有可能导致不可行路径上的信息素过高，引导较多蚂蚁行走不可行路径，蚂蚁陷入“死胡同”，使算法死锁停滞，规划不出合理的路径。IDACA 在复杂的环境下能规划出较优的路径，其原因首先通过改变距离启发因子，增强目标点对蚂蚁的引导作用，使蚂蚁尽量向目标点靠近，权值系数和信息素挥发系数的动态调整避免由于信息

素的决定性作用使得进化后期出现早熟收敛。在图 3-9 中的凹环境中，ACA 规划的路径比 IDACA 和 DACA 规划的路径有更多的尖角，路线更弯曲。其原因是：ACA 在凹形环境下蚂蚁的盲目搜索，致使蚂蚁在凹环境下迂回搜索，较难走出凹环境。导致较多蚂蚁陷入凹形环境中，较多蚂蚁规划不出路径，使进化过程难以进行，蚁群算法依靠信息素正反馈的进化优势没有得到发挥。而 IDACA 可以在凹形环境下规划出较合理的路径，通过改进距离启发因子可以增强目标点对蚂蚁的引导作用，权值系数和挥发系数的动态调整，可以避免蚂蚁的在凹行障碍物周围迂回搜索，较多蚂蚁可以走出凹形障碍物，同时引入路径交叉也可以避免蚂蚁陷入凹形环境。

三种算法得到的路径规划长度比较如表 3-2 所示。

表 3-2 三种算法规划路径时得到的路径长度

测试实例	障碍物环境	路径长度		
		ACA	DACA	IDACA
测试实例 1: 20×20 随机障碍物	$\rho = 0.05$	382.13	267.98	253.81
	$\rho = 0.1$	351.42	259.70	253.84
	$\rho = 0.3$	418.99	315.13	295.56
	$\rho = 0.5$	413.13	298.99	287.28
测试实例 2: 30×30 随机障碍物	$\rho = 0.05$	589.97	448.70	422.84
	$\rho = 0.1$	548.70	422.43	386.27
	$\rho = 0.3$	599.81	492.23	457.98
	$\rho = 0.5$	592.14	485.56	454.31
测试实例 3: 40×40 随机障碍物	$\rho = 0.05$	887.68	678.40	567.69
	$\rho = 0.1$	865.98	657.69	549.41
	$\rho = 0.3$	974.17	767.98	636.27
	$\rho = 0.5$	835.61	638.70	582.13

从表（3-2）中的数据可以得出如图 3-10 所示的曲线图。

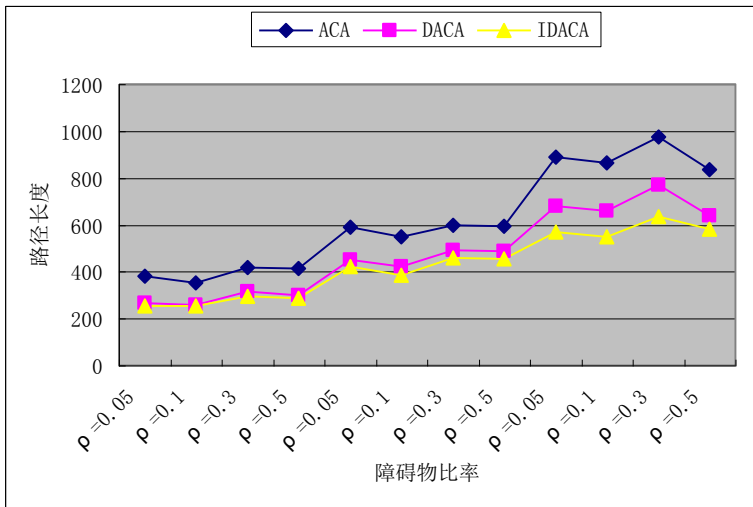


图 3-10 三种算法在不同环境障碍物比率下规划路径长度对比

根据表(3-2)中的测试实例 1, 障碍物比率为 0.05、0.1、0.3、0.5 时, IDACA 规划的路径长度比 ACA 规划的路径长度分别优化提高了 33.6%、27.7%、29.4%、30.5%, IDACA 规划的路径长度比 DACA 规划的路径长度优化提高了分别为 5.3%、2.4%、6.2%、3.9%。根据表(3-2)中的实例 2, 障碍物比率为 0.05、0.1、0.3、0.5 时, IDACA 规划的路径长度较 ACA 规划的路径长度优化提高了 28.3%、29.6%、23.6%, IDACA 规划的路径长度较 DACA 规划的路径长度优化提高了 5.8%、8.7%、7.0%、6.3%, 根据表(3-2)中的实例 3, 障碍物比率为 0.05、0.1、0.3、0.5 时, IDACA 规划的路径长度较 ACA 规划的路径长度分别优化提高了 36.1%、36.3%、34.7%、30.3%。IDACA 较 DACA 规划的路径长度分别优化提高了 16.4%、16.5%、17.2%、8.9%。从以上的数据分析和图(3-10)中可以看出 IDACA 较 ACA 和 DACA 规划的路径更短, 得到结果更理想。从以上数据分析中也可以看出在复杂环境下 IDACA 规划的路径较 ACA 和 DACA 规划的路径优化效果更为明显。

3.7 本章小结

本章首先介绍了路径规划的环境和问题描述, 其次介绍了蚁群算法路径规划的基本原理, 然后介绍了本文提出的四点改进策略, 最后通过实验仿真同基本蚁群算法和文献[58]提出的蚁群算法进行比较和验证, 以此说明本文提出的改进策略的合理性和有效性。本章并没有涉及到蚂蚁如何避免陷入“死胡同”, 且改进的双蚁群算法用于机器人路径规划的时间较传统蚁群算法用于机器人路径规划的时间稍长, 针对这些缺点和不足, 下一章将提出进一步的改进策略。

第四章 复杂环境下基于改进的蚁群算法机器人路径规划

第三章提出的改进的双蚁群算法比传统蚁群算法规划的路径较短，且在复杂的环境下改进的双蚁群算法也可以规划较合理的路径，但传统蚁群算法在复杂的环境下就难以规划出合理的路径，由于蚂蚁盲目搜索，无法避免蚂蚁陷入“死胡同”，导致算法停滞，规划不出合理的路径。第三章提出的改进双蚁群算法也有其不足，在规划时间的耗费上要比传统的蚁群算法稍长，且并没有涉及到机器人在几何形状障碍物环境下的路径规划，针对第三章提出改进算法的缺点和不足，第四章在第三章基础上提出了进一步的改进，通过改变距离启发算子，引入动态更新路径点信息素、基于概率的路径点最优性选择策略和路径点回退策略对蚁群算法进行改进，弥补上一章提出算法的不足。

4.1 基本蚁群算法的原理

蚂蚁在运动时会在通过的路径上释放一种特殊的分泌物——信息素来寻找路径。当它们碰到一个还没有走过的路口时，就随机地挑选一条路径前行，同时释放出与路径有关的信息素。蚂蚁走的路径越长，则释放的信息量越小。当后来的蚂蚁再次碰到这个路口的时候，选择信息量较大的路径概率相对较大，这样便形成了一个正反馈机制。最优路径上的信息量越来越大，而其他路径上的信息量却会随时间的流逝消减，最终整个蚁群会找出最优路径。

4.1.1 节点选择规则

蚂蚁K在运动过程中，会根据每条路径上的信息素浓度决定下一步的转移方向，在t时刻，如图4-1设蚂蚁k在点A(i,j)处，待选节点B(k,l),t时刻蚂蚁由位置A转移到位置B的概率为；

$$p_{ab}^k(t) = \begin{cases} \frac{\tau_{ab}^\alpha(t)\eta_{ab}^\beta(t)}{\sum_{s \in M_{ab(i,j)}^k} \tau_{as}^\alpha(t)\eta_{as}^\beta(t)} & s \in M_{ab(i,j)}^k \\ 0 & s \notin M_{ab(i,j)}^k \end{cases} \quad (4-1)$$

式中 η_{ab} 为线段 AB 的能见度，为点 A 和点 B 之间距离的倒数，即 $1/d_{ij}$ ； τ_{ab} 为线段 AB 上的信息素浓度， α 为信息激素物质的相对重要性； β 为能见度的相对重要性， $M_{ab(i,j)}^k$ 为当前点 A 下一步待选节点的集合。

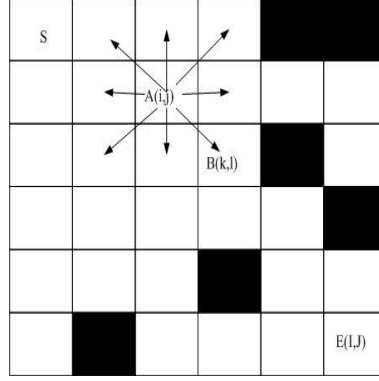


图4-1环境栅格模型

4.1.2 信息素更新规则

在每次循环中，每只蚂蚁应用信息素更新规则，只对走过的节点进行更新，其他节点的信息素只是进行信息素挥发。信息素更新规则如式(4-2)所示。

$$\tau(j) \leftarrow (1 - \rho)\tau(j) + \Delta\tau(j) \quad (4-2)$$

在式(4-2)中 ρ 为信息素挥发因子 ($0 < \rho < 1$)； $\Delta\tau(j)$ 为更新因子，如式(4-3)所示。

$$\Delta\tau(j) = \sum_{k=1}^m \Delta\tau_k(j) \quad (4-3)$$

其中：

$$\Delta\tau^k(j) = \begin{cases} \frac{Q}{d_{ij}} & \text{蚂蚁K经过节点j} \\ 0 & \text{蚂蚁k不经过节点j} \end{cases} \quad (4-4)$$

式(4-4)中， Q 为信息量常数， d_{ij} 为蚂蚁 k 在本次循环中当前点到其邻域的距离。

4.2 蚁群算法改进策略

4.2.1 距离启发因子的改进

式子 (4-1) 中 η_{ab} 作为距离启发因子，传统的蚁群算法中距离启发因子为 i

和 j 之间的距离倒数 $1/d_{ij}$ 。在栅格的环境模型中，相邻栅格的中心之间的距离为 1 或者为 $\sqrt{2}$ ，差别很小，对蚂蚁选择下一个节点的启发性不明显，增大蚂蚁搜索下一个节点的盲目性，得不到最优路径。为了使蚂蚁向离目标点较近的点选择，增强距离启发因子对蚂蚁的选择下一个节点的启发性，本文提出一种新的距离启发因子，如式 (4-5) 所示。

$$\eta_B = 1 - \frac{d_{AB} + d_{BE}}{\sum_{C \in M_{A(i,j)}} d_{AC} + d_{CE}} \quad (4-5)$$

如图 4-1, 式 (4-5) 中 d_{AB} 表示当前蚂蚁所在的栅格点 A 到其邻域栅格点 B 的距离， d_{BE} 表示当前栅格点的邻域栅格点 B 到目标栅格点 E 的距离。 $M_{A(i,j)}$ 为当前栅格点 A 到下一步可行栅格点的集合。如果当前栅格点 A 到其邻域的距离加上到其邻域到目标栅格点的距离较大，即当前被选择栅格点离目标点较远，距离启发因子较小，当前被选择的栅格概率就较小。反之当前被选栅格更靠近目标点，距离启发因子就更大，被选择的概率就更大，启发蚂蚁向靠近目标点的路径行走。在大规模障碍物稀疏的环境中此种距离启发因子的改进，减少了蚂蚁盲目大规模搜索，使蚂蚁向靠近目标点的区域搜索，减少搜索时间和空间，规划出的路径更优。

4.2.2 基于排序的最优性选择下一个路径栅格

传统蚁群算法为了加强蚂蚁当前栅格选择下一个栅格随机性，采用了“轮盘赌”方法来选择下个可行栅格节点。采用“轮盘赌”方式中，随机数的产生会影响“轮盘赌”对下个节点的选择，合理的随机数可以使“轮盘赌”方式选择合理的节点，不合理的随机数会严重影响到“轮盘赌”对下个节点的选择，有可能使蚂蚁得到的路径是局部最优，或者大部分蚂蚁陷入死胡同，得不到合理的路径。传统蚁群中采用的“轮盘赌”，并没有对被选栅格点进行排序，可能导致有的栅格永远不能被选择到，使其得到的路径未必是全局的最优路径。针对此缺点本文采用基于排序的最优性选择策略，对被选择栅格按概率进行升序排列，通过轮盘赌每次选择概率最大的栅格点，作为蚂蚁下一个选择的栅格点，减少蚂蚁对路径的选择的随机性。

4.2.3 信息素初始化的改进

传统的蚁群算法，将蚂蚁行走空间中每条路径都初始化为一定的信息素。这样就导致刚开始蚂蚁选择下个路径点时，信息素启发因子启发的效果并不明显，

使蚂蚁盲目搜索路径，致使算法性能降低。为此，本文对之进行如下改进：

(1) 初始化时，蚂蚁的行走空间信息素初始化为 0；

(2) 蚂蚁刚开始对周围下个邻域点选择时，路径点上信息素设置为距离启发因子的固定倍数。距离启发因子越大，该点上信息素加的越多，节点被选择的相对概率就越大，反之，该点路径上的信息加的越少，被选择的概率就越小。

通过上述改进，使信息素启发因子发挥明显的启发效果。利用 MMAS 思想，对路径上的信息素加以控制，防止蚂蚁出现早熟收敛，陷入局部最优。

4.2.4 不可行路径信息更新规则

传统蚁群算法对不可行路径的更新采用传统的局部更新或者采用全局更新方式。采用局部更新，蚂蚁每走一步就更新走过的节点信息素，如果此路径不可行且路径节点信息素过高，会引导大多数蚂蚁集中行走此路径，一旦部分蚂蚁陷入死胡同，将导致算法进化后期大多数蚂蚁走入死胡同，得不出合理路径。采用全局更新，蚂蚁只能更新可行路径节点的信息素，不可行节点的信息素只进行挥发。如果此路径很少蚂蚁走过，其信息素过少，很可能大多数蚂蚁不会选择此路径节点，减少了蚂蚁搜索路径的范围，规划的路径为局部最优。本文提出一种对不可行路径的动态更新方式，如式 (4-6) 所示。

$$P' = \begin{cases} P + q & (0 < P \leq Q_1) \\ \rho_1 P & (Q_1 < P \leq Q_2) \\ \rho_2 P & (Q_2 < P \leq Q_3) \\ P - q & (P > Q_3) \end{cases} \quad (4-6)$$

式(4-6)中 P' 表示更新以后当前不可行路径节点信息素， q 为固定的信息素量， Q_1 、 Q_2 、 Q_3 为信息素量， p 为当前不可行路径节点信息素， ρ_1 和 ρ_2 为挥发系数，且 $\rho_1 > \rho_2$ 。如果当前不可行路径节点信息过小，且小于 Q_1 ，则不可行路径节点添加上固定量的信息素 q ；如果不可行路径节点的信息素过高，且高于 Q_3 ，则此路径上节点以固定的量信息素 q 挥发。除此之外，不可行路径节点的信息素按照挥发系数 ρ_1 、 ρ_2 进行挥发。此种方式的改进，可以防止不可行路径节点信息素浓度过高，过多的蚂蚁走入不可行路径中，使其陷入死胡同，得不到理想的路径。由于不可行路径某些节点信息素过少，此不可行路径上的节点有可能使全局路径最优路径上的节点，使蚂蚁可能搜索不到，基于此为了防止此路径上某些节点信息素过少，适当的加上一定量的信息素，增大蚂蚁的全局搜索能力，尽可能找到更优的路径。

4.3 蚂蚁回退策略

蚂蚁在特定的地形下，蚂蚁 K 会遇到后续节点不可行，此时 $|Z_i(k)|=0$, $|Z_i(k)|$ 表示蚂蚁 K 在该时刻可行节点的集合中元素的数目。即蚂蚁陷入“死胡同”，从而算法出现死锁停滞的状况。如图 4-2 所示

0	1	2	3	4
5		7		9
10		12		14
15				19

图 4-2 蚂蚁陷入死胡同示意图

如图 4-2 所示，蚂蚁 K 从起点 0 出发，经过的栅格为 0-1-2-7-12，可以看出此时蚂蚁 K 没有后续节点可以选择，即蚂蚁 K 陷入死胡同，此时 $|Z_i(k)|=0$ 。 $|Z_i(k)|$ 的值是判断蚂蚁 K 在某时刻是否陷入陷阱的标志。

当蚂蚁 K 落入陷阱时，如果没有好的策略使蚂蚁走出陷阱，该蚂蚁将会处于“死亡”状态，导致整个算法处于死锁停滞，使得算法对环境的复杂度和健壮性不够强。

对于陷阱问题，较常用的方法是：在环境初始化，对障碍物做特定的处理，使得环境中所描述的障碍物都是凸的形状，以消除凹形障碍物形成的陷阱。将图 4-2 的栅格环境按照这种方法处理后的结果为图 4-3 所示。

0	1	2	3	4
5				9
10				14
15				19

图 4-3 死胡同处理示意图

0	1	2	3	4

图 4-4 死胡同处理产生示意图

通过这种方法处理能起到较好的作用，但通过这种方法处理以后，虽然消除了单个障碍物生成的陷阱，并没有消除障碍物与障碍物之间或者障碍物与环境边界之间形成的陷阱如 (5-10-15) 和 (9-14-19)，如图 4-4 所示，所以此算存在很大

的局限性。本文提采用一种蚂蚁回退策略，结合图 4-2 表述如下：

- 1) 蚂蚁 K 处于节点 12 时， $|Z_{12}(k)|=0$ ，此时蚂蚁无后续节点可选，蚂蚁 K 陷入陷阱。此时标记节点 12 为不可行节点。 $\text{tabu}(N(K))=12, N(K)$ 表示第 K 只蚂蚁已经走过的步数；
- 2) 蚂蚁 K 从位置 12 回退到 7， $N(K)=N(K)+1, \text{tabu}(N(K))=7$ ；
- 3) 蚂蚁 K 在集合 $(Z_7(K)-\{12\})$ 中重新选择节点，蚂蚁 K 发现 $|Z_7(K)-\{12\}|=0$ ，即蚂蚁又陷入陷阱，将节点 7 标记为不可行节点；
- 4) 蚂蚁 K 再从 7 回退到 2， $N(K)=N(K)+1, \text{tabu}(N(K))=2$ ；
- 5) 蚂蚁 K 在集合 $(Z_2(K)-\{7\})$ 中重新选择节点，蚂蚁发现 $|Z_2(K)-\{7\}| \neq 0$ ，此时蚂蚁可选择后续节点 3，蚂蚁 K 跳出了陷阱。

回退策略的示意图如图 4-5 所示。

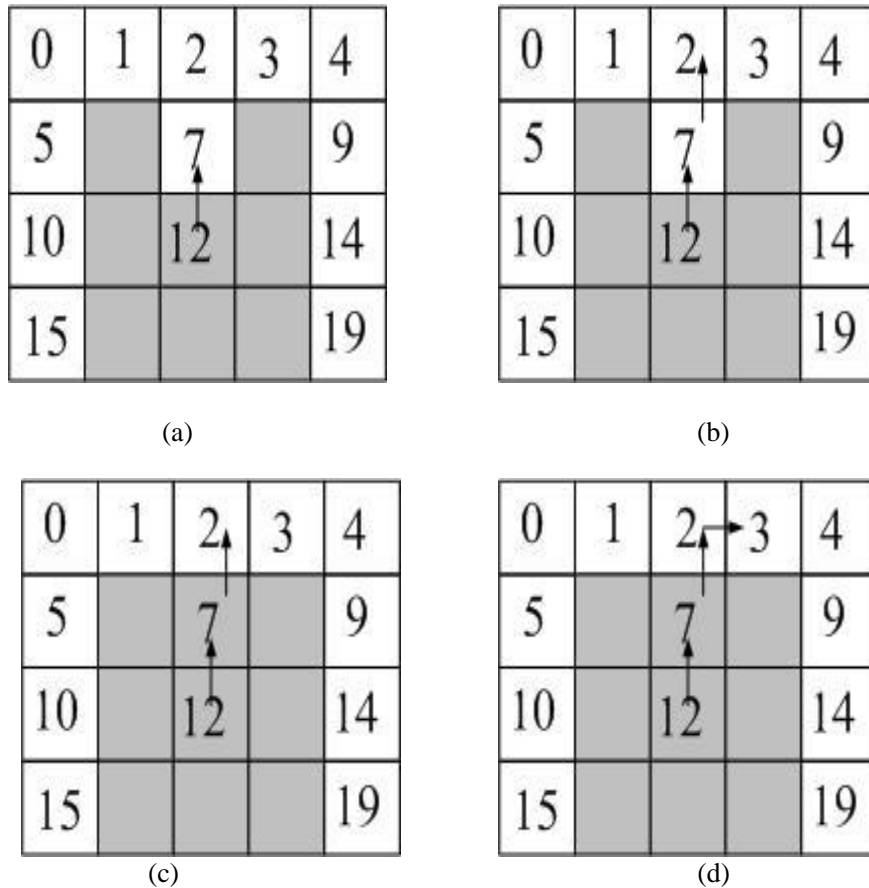


图 4-5 蚂蚁回退示意图

4.4 改进算法步骤

Step1: 首先建立机器人环境栅格模型，对每一个栅格节点初始化信息素为 0，设置蚂蚁数为 M，最大循环次数为 Nmax，最大步数为 steplimit，最小信息素为 τ_{\min} ，

最大信息素为 τ_{\max} ，初始化循环次数 $Nm=0$ ，初始化其他参数，转 **Step2**;

Step2:在起点放置 M 只蚂蚁，对每只蚂蚁按照式(4-1)中的概率 P ，利用确定性概率策略选择进行下个节点的选择，把所得节点放到每只蚂蚁的禁忌表中，转

Step3;

Step3:判断当前节点周围是否有可行栅格点，如果周围没有可行栅格点，将该点从禁忌表删除;转 **step2**;如果当前栅格点有可行节点，转 **step4**;

Step4 每只蚂蚁在对其走过的节点，按照式((4-2)~(4.4))进行信息素更新，如果更新后的当前节点信息素小于 τ_{\min} ，则将当前节点的信息素为 τ_{\min} ，如果更新后的当前节点信息素大于 τ_{\max} ，则当前节点信息素为 τ_{\max} ，转 **Step5**;

Step5:循环执行 **step2-step4**，直到所有蚂蚁都到达目标点，对其陷入“死胡同”的蚂蚁走过的不可行路径按照式(4-6)不可行路径信息素更新规则进行更新， $Nm++$ ，转 **Step6**;

Step6:当 $Nm \leq N_{\max}$ 时，循环执行 **step2~step5** 直到得到最优解满足设定要求。

4.5 仿真结果分析

4.5.1 仿真实例和参数设定

实例 1：随机模型，栅格数为 20×20 ，障碍物比率 $p=0.05, 0.1, 0.3, 0.5$;

实例 2：随机模型，栅格数为 30×30 ，障碍物比率 $p=0.05, 0.1, 0.3, 0.5$;

实例 3：凹形障碍模型，栅格数分别为 20×20 、 30×30 ;

实例 4：4 种规则的稀疏障碍模型（Wall、Circle、H 型、S 型），栅格数为 40×40 。

上述 5 种实例的起点、目标点和环境空间的配置分别如图 4-6~4-9 所示，栅格大小为 10。仿真参数如下表 4-1

表 4-1 仿真参数表

α	β	ρ	M	N_{\max}	steplimit	q	$Q1$	$Q2$	$Q3$	$\rho1$	$\rho2$	τ_{\min}	τ_{\max}
2.0	2.5	0.7	20	300	250	10	30	70	150	0.7	0.2	20	400

4.5.2 仿真结果与性能比较

为了验证性能，对上述 5 种实例进行测试，在 Windows XP 系统下(CPU 2.0GHZ,

内存 1GB)VC++ 6.0 开发环境下,分析比较本文所提出改进算法与蚁群算法(ACA: Ant Colony Algorithm)、第三章提出的改进的双蚁群算法的性能。为表述方便,将本文算法简称为 IACA: (Improved Ant Colony Algorithm), 第三章提出的改进的双蚁群算法简称为: IDACA(Improve Double Ant Colony Algorithm),三种算法路径规划的路径如图(4-6~4-9)所示, 图中蓝色表示蚁群算法 (ACA), 紫色表示改进的蚁群算法(IACA),绿色为第三章提出的双蚁群算法(IDACA)。

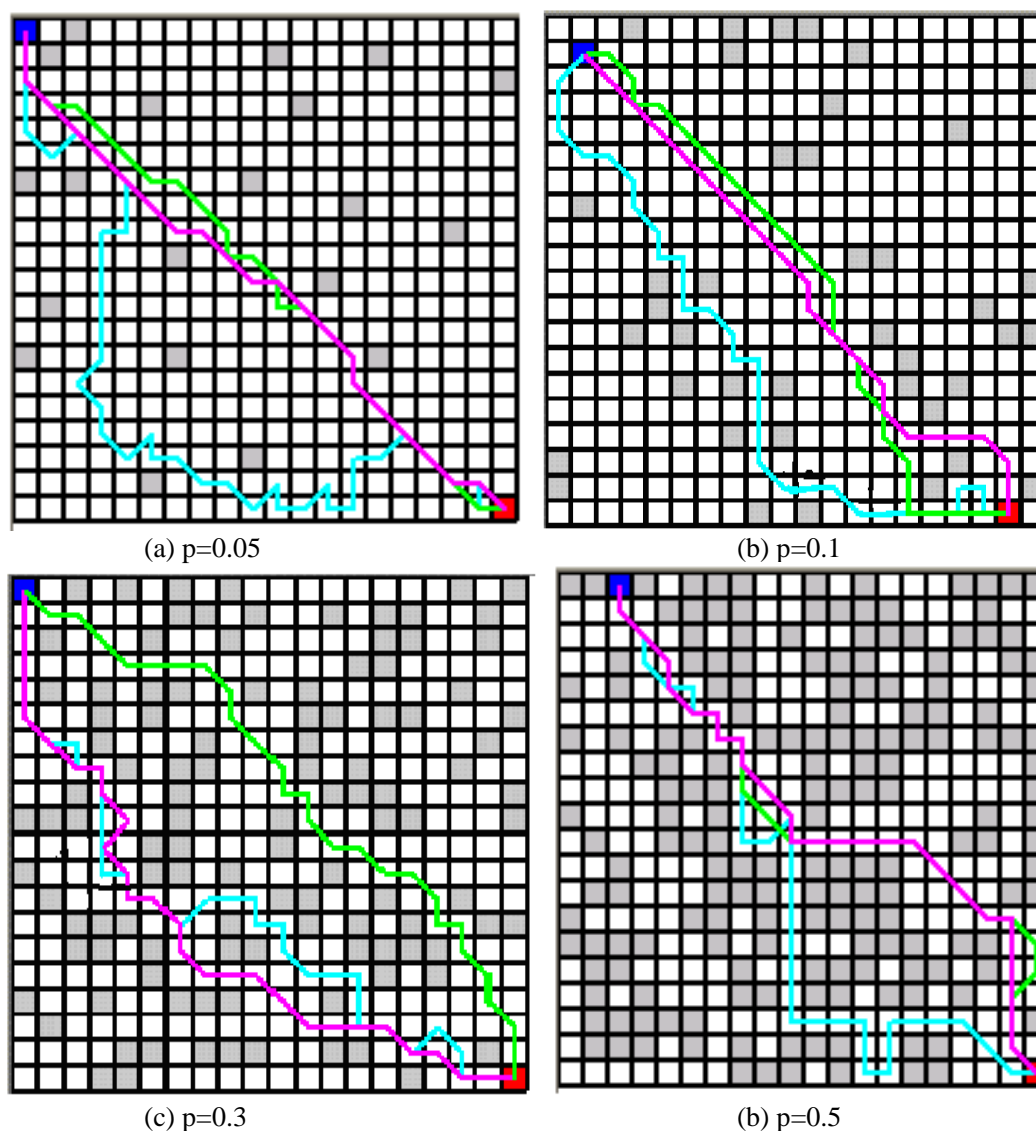


图 4-6 仿真区域为 20×20 栅格时, 随机障碍物环境下的仿真结果比较 (p 为障碍物比例)

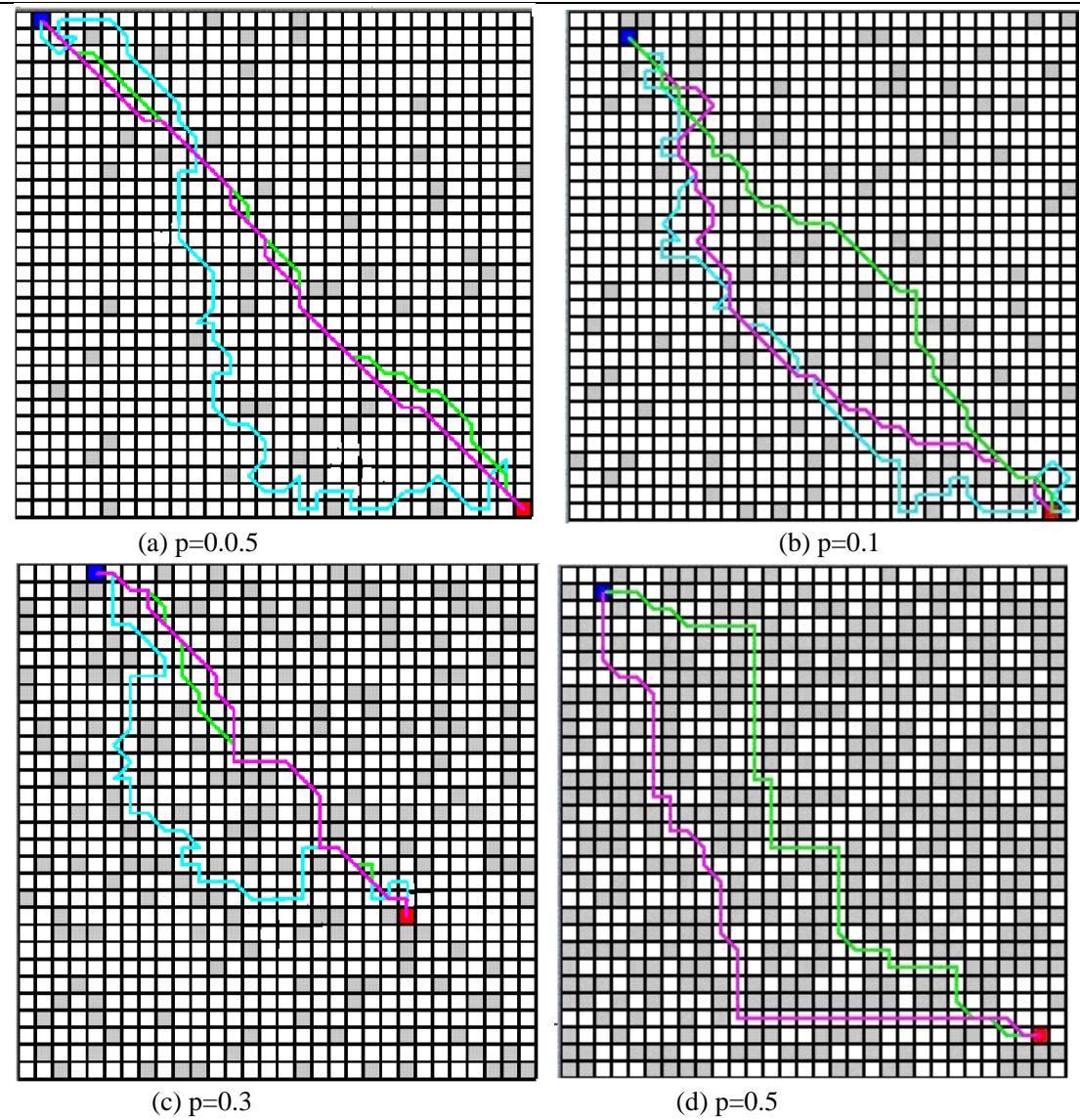


图 4-7 仿真区域为 30×30 栅格时，随机障碍物环境下的仿真结果比较（ p 为障碍物比例）

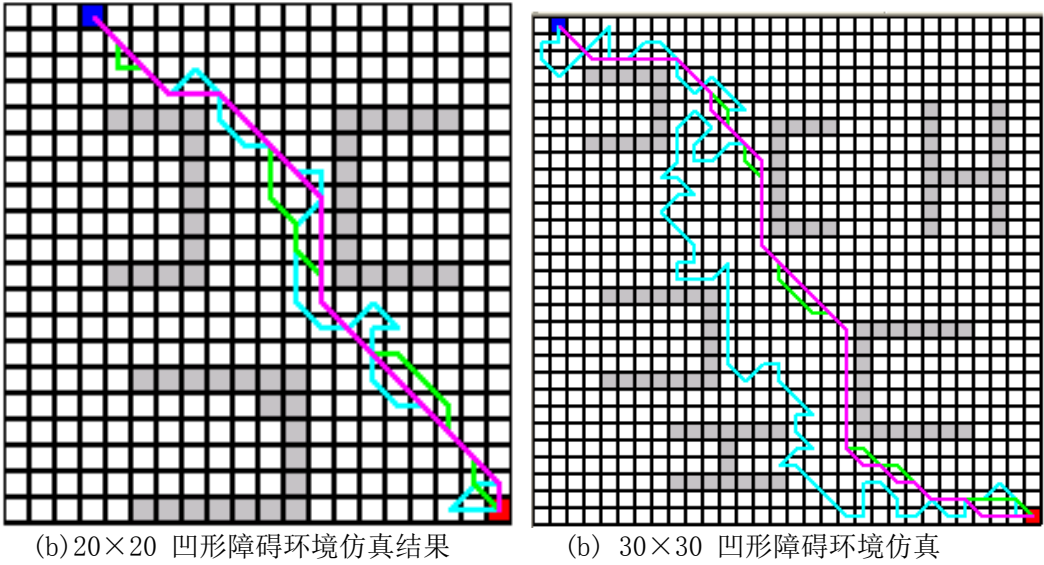


图 4-8 凹形障碍物仿真结果

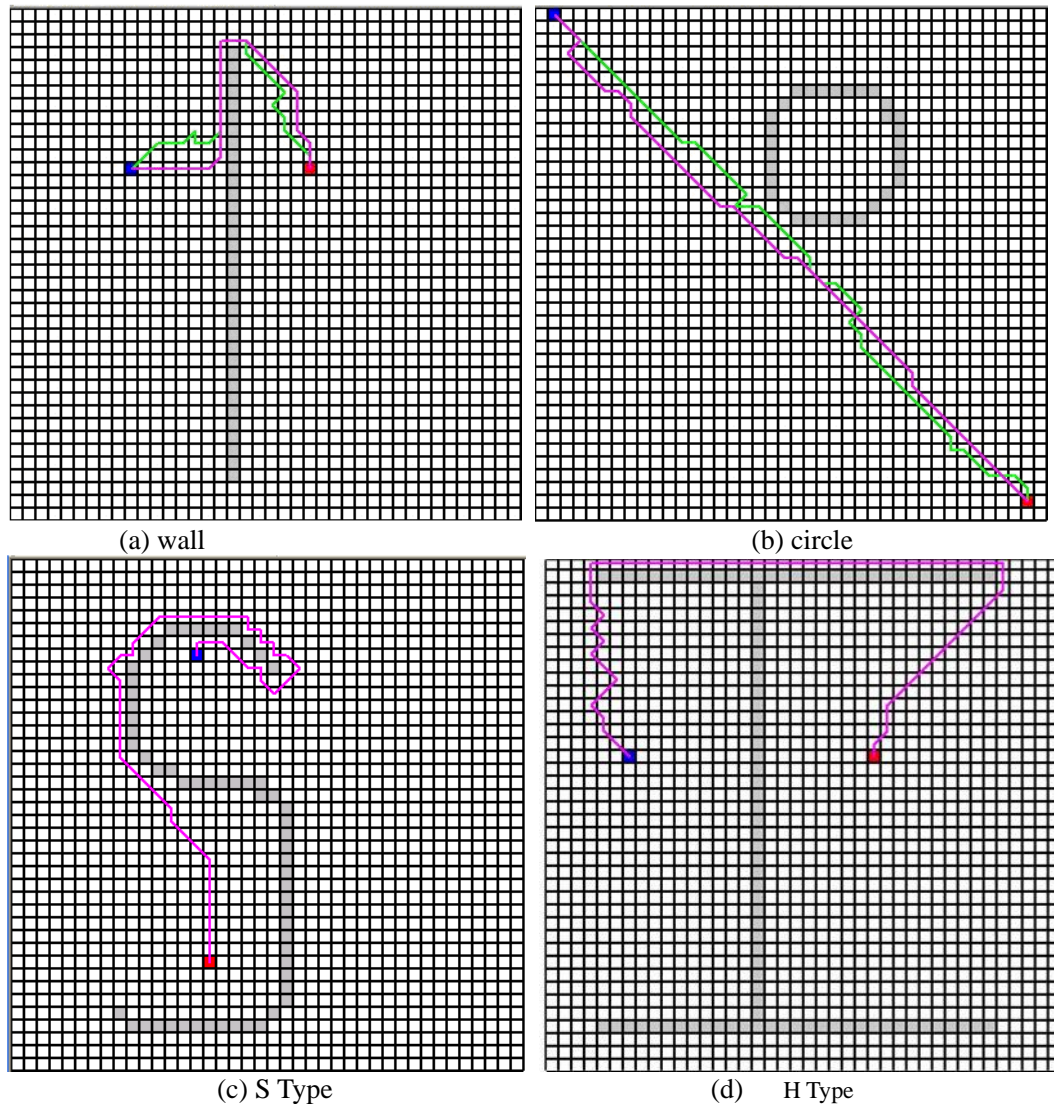


图 4-9 仿真区域为 40×40 栅格时，规则稀疏障碍物环境下 IACA 仿真结果

从图 4-6~4-8 可以看出 ACA 规划出的路径长度明显长于 IACA 规划出的路径，且 ACA 规划出的路径存在较多的尖角，路径较曲折。IDACA 规划的路径虽然比 ACA 规划的路径要优化的多，且与 IACA 规划的路径有部分重合，但 IDACA 规划的路径也有多处存在弯折，规划的路径也比 IACA 规划的路径稍长。IACA 规划出的路径更平滑，尖角较少，规划的路径更优。在图 4-7 中的(d)图 ACA 在复杂障碍物环境规划不出合理的路径，而 IACA 可以规划出较理想的路径。其原因是：在复杂环境下，蚂蚁可行路径较少，在开始阶段部分蚂蚁走入“死胡同”，使部分蚂蚁走入不可行路径，蚂蚁在不可行路径上留下信息素，使其不可行路径信息素过高，引导后期较多蚂蚁走其先前蚂蚁走过的路径点，使后期较多蚂蚁陷入“死胡同”，导致绝大部分蚂蚁走入“死胡同”，算法停滞，规划不出合理的路径。IACA

通过对蚂蚁不可行的路径上信息素采用分段动态更新,使其不可行路径信息素不会过高,大部分蚂蚁不会被再次引导陷入“死胡同”,同时采用回退策略,使蚂蚁陷入“死胡同”时,可以回退继续搜索可行节点,使其规划出较优的路径。图 4-9 中是 IACA 和 IDACA 在四种障碍物模型下规划出的结果,ACA 在这四种障碍物下规划不出理想的路径。其原因:(1)ACA 中距离启发因子对蚂蚁的启发并没有明显的启发作用,在这四种模型中,起点和终点之间蚂蚁可搜索的范围比较广,启发因子作用不明显使蚂蚁盲目随机搜索,不会启发蚂蚁向靠近终点的路径点选择,而 IACA 通过改进的距离启发因子,使蚂蚁尽量选择靠近终点的路径点选择,不会盲目在可视区域搜索。(2)ACA 在起点的可视区域会出现迂回,回到起点周围的区域,致使蚂蚁走不出该区域,导致起点周围区域信息素浓度较高,致使全部蚂蚁在信息素的启发下,在其在起点周围寻找路径,使全部蚂蚁最终找不到路径。IDACA 在 S 模型和 H 模型障碍下,得不到合理的路径。原因是:在 S 模型和 H 模型环境下,大部分蚂蚁在可视的范围区域迂回搜索,致使蚂蚁走不出起点周围的区域,导致 IDACA 在 S 模型和 H 模型障碍物环境中规划不出理想的路径。IACA 对不可行路径进行信息素调整,使绝大部分蚂蚁可以走出该区域,找到较优的路径。针对此三种算法在路径长度和计算机时间上的差异做如下表 4-2 所示的比较。

表 4-2 三种算法规划路径时得到的路径长度与计算时间比较

测试实例	障碍物环境	路径长度			时间(s)		
		ACA	IDACA	IACA	ACA	IDACA	IACA
测试实例 1: 20 × 20 随机障碍物	$\rho = 0.05$	402.13	279.70	257.98	16.13	16.34	15.01
	$\rho = 0.1$	437.98	277.98	255.56	15.61	15.84	14.51
	$\rho = 0.3$	487.28	338.99	316.58	10.98	11.20	9.87
	$\rho = 0.5$	413.14	298.64	277.18	5.68	5.87	4.97
测试实例 2: 30 × 30 随机障碍物	$\rho = 0.05$	565.27	420.42	402.84	26.9	27.3	25.4
	$\rho = 0.1$	665.98	461.13	424.56	27.01	27.41	26.1
	$\rho = 0.3$	689.71	492.14	457.78	11.08	11.47	9.87
	$\rho = 0.5$	686.58	490.71	466.27	7.05	6.76	6.02
测试实例 3: 40 × 40 随机障碍物	$\rho = 0.05$	785.98	565.98	530.83	36.1	38.2	34.2
	$\rho = 0.1$	768.41	545.98	515.84	35.9	37.5	33.8
	$\rho = 0.3$	910.21	699.71	646.27	29.71	30.83	27.57
	$\rho = 0.5$	832.74	622.84	596.98	21.56	22.03	19.69

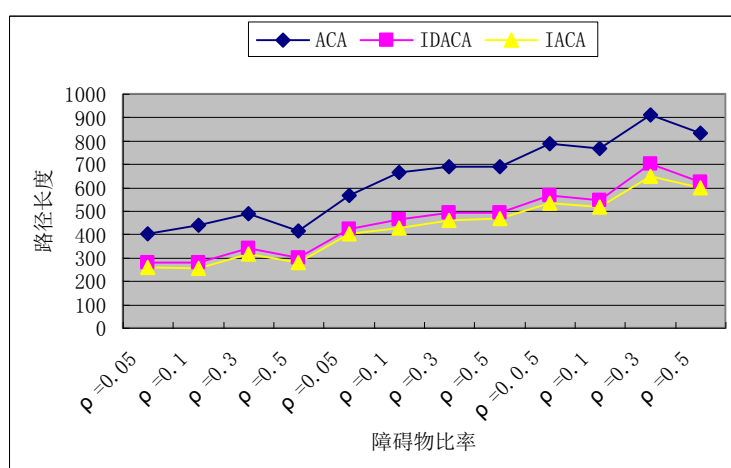


图 4-10 三种算法在不同障碍物比率环境下的路径长度对比

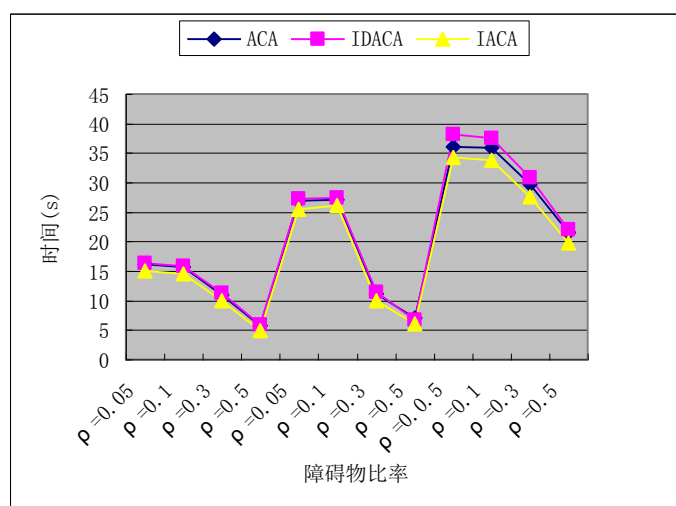


图 4-11 三种算法在不同障碍物比率环境下的规划路径时间对比

根据表(4-2)的测试实例 1, 障碍物比率为 0.05、0.1、0.3、0.5 时, IACA 规划的路径长度比 ACA 规划的路径长度优化提高分别为 35.8%、37.1%、35.0%、30.5%, IACA 规划的路径长度比 IDACA 规划的路径长度优化提高了分别为 7.8%、8.1%、6.6%、7.2%。根据表(4-2) 的测试实例 2, 障碍物比率为 0.05、0.1、0.3、0.5 时, IACA 规划的路径长度比 ACA 规划的路径长度优化提高分别为 28.7%、36.3%、33.6%、32%, 比 IDACA 路径长度优化提高 4.2%、7.9%、7.0%、5.0%, 根据表(4-2) 的测试实例 3, 障碍物比率为 0.05、0.1、0.3、0.5, IACA 规划的路径长度比 ACA 规划的路径长度优化提高为 28%、32.9%、28.9%、28.3%, 在障碍物为 0.05、0.1、0.3 和 0.5 时, IACA 规划的路径长度比 IDACA 规划路径长度优化提高分别为 6.2%、5.5%、7.6%、4.2%。从以上分析可以看出 IACA 比 ACA 在路径长度优化提高在 25%~38%, 且 IACA 比 IDACA 在路径上优化提高在 5%~9%, 从时间上看 IACA

比 ACA 在时间上提高了 5%~15%，在时间上 IACA 比 IDACA 在时间上提高了 5%~15%。第三章提出的 IDACA 算法在时间上比 ACA 上长 1%~5%，但 IDACA 规划的路径比 ACA 规划的路径长度优化提高了 20%~35%。IDACA 在时间上比 ACA 长，但在路径长度得到较大的优化前提下，在时间耗费上长一些是可以接受的，也是合理的。同时，IDACA 在复杂的环境下可以规划出较合理的路径。在从以上分析和图(4-10~4-11)可以看出，IACA 在时间上比 IDACA、ACA 更节省时间，在路径规划长度比 IDACA、ACA 规划的路径更短。

4.6 本章小结

本章通过分析基本蚁群算法和上一章提出改进算法的缺点，提出进一步的改进。通过与上一章提出的改进的蚁群算法和基本蚁群算法的仿真验证的对比，验证了本章改进策略的有效性和可行性。

第五章 总结与展望

5.1 总结

在分析目前各种移动机器人路径规划方法的基础上，本文对蚁群算法用于路径规划问题进行了分析研究。在研究基本蚁群算法的理论基础上，本文做了两种方式的改进，并将它们应用于移动机器人路径规划中。研究的具体工作总结如下：

(1) 本文在单蚁群算法的基础上提出了改进：改进启发因子，使蚂蚁尽量靠近目标点的方向移动，动态调制权值系数和信息素挥发系是为避免算法陷入局部最优，得不到最优路径。

(2) 本文在第三章提出的改进基础上，进一步分析改进的不足，又提出进一步的改进，通过改进距离启发因子，使算法中蚂蚁合理的选择下个节点；不可行路径的信息素动态更新，防止蚂蚁陷入死胡同，算法停滞；初始化信息更新防止蚂蚁避免蚂蚁盲目搜索，基于排序的最优性概率选择，使蚂蚁更容易搜索下个可行栅格节点。

(3) 本文所研究的机器人路径规划方法是以栅格化为环境模型。通过提出的改进策略与传统的蚁群算法和其它改进的蚁群算法进行仿真比较，验证了本文提出的改进策略的可行性和有效性。

5.2 展望

(1) 机器人路径规划中并没考虑到机器人的机械特性，而是将机器人当做质点考虑。机器人在真实的工作环境中，应当考虑机器人的拐角。

(2) 本文研究的路径规划，是在全局信息已知的工作环境中，而实际中的移动机器人的工作环境是动态未知的环境。如果能使本文提出的改进用于动态环境中，采用一定的避障策略，能实时避开障碍物，将很大程度上提高算法的性能，扩展算法的应用。

(3) 本文研究的是在二维平面环境下的路径规划，并没有考虑高度信息。在三维空间环境下实现路径规划，考虑的因素更多，随着维数的增加，路径搜索空间呈指数增加。

(4) 蚁群算法在大规模复杂的凹形环境中难以规划出最优的路径，由于凹形环境空间比较复杂，路径搜索空间较大，导致算法计算量和存储空间较大，规划不出最优的路径。

综上所述，以上四点需要在以后学习和工作中进一步完善。

参考文献

- [1] B Bullnheimer,R F Hartl,C Strauss.A New Rank-based Version of The ant System:A Computation Study[R].Technical Report POM-03/97,Institute of Management Science, Universtiy of Vlenna,1997
- [2] Brooks R A.Solving the Find-Path Problem by good representation of free space[J].IEEE Trans on Sys Man Cybem,1983,13(3):190~197
- [3] 李善寿, 方潜生, 肖本贤等. 全局路径规划中基于改进可视图法的环境建模[J]. 华东交通大学学报. 2008,25(6):73~74
- [4] 刘作军. 基于电路地图的移动机器人路径规划方法研究[D]:[博士学位论文]. 天津南开大学. 2005:8~12
- [5] 朱庆宝, 张玉兰. 基于栅格法的机器人路径规划蚁群算法[J]. 机器人. 2005,27(2):132~136
- [6] 王卫红, 顾国民等. 基于栅格法的矢量路径规划算法[J]. 计算机应用研究. 2006,23(3):57~59
- [7] 袁增任, 高明. 在动态环境中移动机器人导航与避障的一种新方法 [J]. 机器人. 2000,22(2):81~88
- [8] 杨淮清, 肖兴贵, 姚栋. 一种基于可视图法的机器人全局路径规划算法[J]. 沈阳工业大学学报. 2009, 31(2):225~228
- [9] Nilsson N J.Artificial Intelligence-A New Synthesis. San Mateo:Morgan Kauffman Publisher.Inc.1998
- [10] Al-Hasan S,Vachtsevanos G.Intelligent route planning for fast autonomous vehicles operating in a large natural terrain.Robotic and Automous Systems[J],2002,40(1):1~24
- [11] Szczerba RJ.Threat Netting for Real-Time,Intelligent Route Planners[C]//The Proceedings of the Information,Decision,and Control Symposium.Adelaide,Australia,1999:377-382
- [12] Szczerba R J, Galkowski P, Glickstein I S,et al.Robust algorithm for real-time route planning[J].IEEE Transactions on Aerospace and Electronic System,2000,36(3):869~878
- [13] 李春华,郑昌文,周成平等.一种三维航迹快速搜索方法[J].宇航学报.2002,23(3):13~17
- [14] 李春华.飞行器三维航迹规划算法研究[D].武汉: 华中科技大学图像识别与人工智能研究所, 2003
- [15] Stentz. A.optimal and efficient route planning for partially-known environments[C].in Proceeding of IEEE Conference on Robotics and Automation,1994:3310~3317
- [16] Stentz A. The focused D* algorithm for real-time replanning[C] .in proceedings of the International Joint Conference on Artificial Intelligence,1995
- [17] O.Khatib.Real time obstacle avoidance for manipulators and mobile robots[J].International

- Journal of Robotics Research.1986,Vol.5(1):90~99
- [18] 庄晓东, 孟庆春, 高云等. 复杂环境中基于人工势场优化算法的最优路径规划[J]. 机器人. 2003,25(6):531~535
- [19] 石为人, 黄兴华, 周伟. 基于改进人工势场法的移动机器人路径规划[J]. 计算机应用研究. 2010,30(8):2021~2023
- [20] 张建英, 赵志萍等. 基于人工势场法的机器人路径规划[J]. 哈尔滨工业大学学报. 2006,38(8):1306~1309
- [21] 张建英, 刘敦. 基于人工势场法的移动机器人最优路径规划[J]. 航空学报.2007,28,183~188
- [22] 王仲民, 岳宏. 一种移动机器人全局路径规划新型算法[J]. 机器人. 2003, 25 (2) : 152~155.
- [23] Nakono R.Yamada T.Coventional Genetic Algorithm for Job Shop Problems Proc of 4th,Int Conf on Genetic Algorithms and Their Applications,1991,474~479
- [24] 李庆中, 顾伟康, 叶秀清. 基于遗传算法的移动机器人动态避障路径规划方法[J]. 模式识别与人工智能. 2002,15(2):161~165
- [25] 王强, 姚进, 王进戈. 基于遗传算法的移动机器人的一种路径规划方法[J]. 哈尔滨工业大学学报. 2004,36(7):867~870
- [26] 周兰凤, 洪炳熔. 用基于知识的遗传算法实现移动机器人路径规划[J]. 电子学报. 2006,34(5):911~914
- [27] Kennedy J, Ebethart R. Particle Swarm Optimization[C]. Proceeding of 1995 IEEE international Conference on Neural Networks, New York ,USA,IEEE,1995:1942~1948
- [28] 秦元庆, 孙德宝, 李宁等. 基于粒子群算法移动机器人路径规划[J]. 机器人.2004,26(3):222~225
- [29] 谭冠政, 刘关俊. 基于粒子群算法的移动机器人全局最优路径规划[J]. 计算机应用研究.2007,24(11):210~212
- [30] 孙波, 陈卫东, 席裕庚. 基于粒子群优化算法的移动机器人全局路径规划[J]. 控制与决策.2005,20(9):1052~1055
- [31] Marco Dorigo,Thomas Stutzle. 蚁群优化. 张军, 胡晓敏, 罗旭耀等. 清华大学出版社, 2007
- [32] M.Dorigo,DiCaro,L.M.Gambardella. Ant algorithms for discrete optimization[R]. Technical ReportIRIDA/98-10.Artif.Life,1999,5(2):137~172
- [33] M.Dorigo and T.Stutzle,Ant Colony Optimization.Cambridge,MA:MIT Press,2004
- [34] M.Dorigo and M.Maniezzo.The Ant System:optimization by a colony of cooperating agents[J].IEEE Transactions on Systems,Man,and Cybernetics-Part B,1996,26(1):29~41P.
- [35] M. Dorigo,V. Maniezzo,A.Colomi.Positive Feedback as a Search Strategy[R].Technical Report No.91-016,Politecnico di Milano,Italy.1991
- [36] A.Colomi,M.Dorigo,V.Maniezzo.Distributed Optimization by Ant Colonies[C].Proceedings of the 1st European Conference on Artificial Life.1991:134~142
- [37] M.Dorigo,V.Maniezzo,A.Colomi.Ant System:Optimization by a Colony of Cooperating

- Agents[J].IEEE Transaction on Systems,Man,and Cybernetics-PartB.1996,26(1):29~41
- [38] M.Dorigo.The Ant Colony Optimization Meta-Heuristic appears in new ideas in Optimization.1999:11~32
- [39] M. Dorigo,T.Stuetzle.The Ant Colony Optimization Metaheuristic Algorithms Applications and Advances[R].Technical Report IRIDIA/2000-32,IRIDIA,Universite Libre de Bruxelles, Belgium. 2000
- [40] Thomas Stitzle,Holger H.Hoos.MAX-MIN ant system and Local Search for Combinatorial Optimization Problem.Future Generation Computer Systems[J].1999:313~329
- [41] Thomas Stitzle,H.H.Hoos.MAX-MIN Ant System.Future Generation Computer Systems[J]. 2000,16(8):889~914
- [42] Walter J.Gutjahr.A Generalized Convergence Result for the Graph Based Ant System Metaheuristic.Engineering and Informational Sciences[J].2003,(17):545~569
- [43] Walter J.Gutjahr.A Converging ACO Algorithm for Stochastic Combinatorial Optimization[J].IEEE Transactions on Evolutionary Computation.2003,(9):10~25
- [44] Thomas Stutzle,Marco Dorigo.A Short Convergence Proof for a Class of ACO Algorithms[J].IEEE Transactions on Evolutionary Computation.2002,6(4):358~365
- [45] Walter J.Gutjahr.A Graph Based Ant System and its Convergence.Future Generation Computer Systems[J].2000,(16):873~888
- [46] 丁建立,陈增强,袁著社.遗传算法与蚂蚁算法的融合[J].计算机研究与发展. 2003,40(9):1351~1356
- [47] 吴斌,史忠植.一种基于蚂蚁算法的 TSP 问题分段求解算法[J].计算机学报. 2001,24(12):1328~1333
- [48] 攀晓平,罗熊,易晟等.复杂环境下基于蚁群优化算法的机器人路径规划[J].控制与决策. 2004,19(2):166~170
- [49] 谢园园,朱庆宝.动态环境下基于蚁群算法的机器人路径规划[J].南京师范大学学报(工程技术版). 2006,6(3):45~50
- [50] 段海滨.蚁群算法原理及应用[M].北京:科学出版社. 2005
- [51] B Bullnheimer,R F Hartl,C Strauss. A New Rank-based Version of The ant System:A Computational Study[R].Technical Report POM-03/97,Institute of Management Science, University of Vienna,1997.Accepted for Publication in the Central European Journal for Operations Research and Economics
- [52] M Dorigo,L M Gambardella.Ant Colony System:A Cooperative Learning Approach to the Traveling Salesman Problem[J].IEEE Trans.On Evolutionary Computation, 1997 ,1(1):53~66
- [53] 李士勇.蚁群算法及其应用[M].哈尔滨;哈尔滨工业大学出版社. 2004
- [54] T.Stitzie,M.Dorigo.ACO Algorithms for the Quadratic Assignment Problem.New

-
- methods in optimization.Mc Graw-Hill.1999:55~66
- [55] A.Colorni,M.Dorigo.Ant System for Job-shop Scheduling. Belgian journal of operations research[J]. Statistics and Computer Science.1994(34):39~53
- [56] 朱庆保. 动态复杂环境下的机器人路径规划蚂蚁预测算法[J]. 计算机学报. 2005,28(11):1898~1906
- [57] 段海滨, 王道波, 朱家强. 蚁群算法理论及应用研究的进展[J]. 控制与决策. 2004,19(2):1321~1326
- [58] 孙纯哲, 林巨广, 楼赣菲等. 凹形障碍物全局路径规划的双蚁群完全交叉算法[J]. 农业机械学报. 2008,39(7):149~153
- [59] T Stutzle,H Hoos.Improvements on the Ant System; Introducing MAX-MIN ant System[C].In Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms, Springer Verlag, Wien.1997.245~249

攻读硕士学位期间论文发表和参加科研的项目情况

- [1] 华路, 周之平。不确定环境下一种新的双蚁群路径规划算法. 计算机工程与应用, 2011, 47(15)
- [2] 周之平, 华路。复杂环境路径规划的改进蚁群算法. 计算机工程与设计, 2011, 32(5)
- [3] 周之平, 华路。复杂环境下基于改进的蚁群算法机器人路径规划. 机器人(已投稿)
- [4] 周之平, 黎明, 华路。基于几何相交测试的机器人路径规划算法的研究. 控制与决策 (已录用)

致谢

首先，在此我要对我的导师周之平表示衷心的感谢。在研究生阶段导师周之平给我提供了优良的科研环境，良好的学术氛围。从中我学会了如何做学术研究，怎么发现问题，解决问题。在周之平老师的悉心指导下，我顺利地完成了论文的编写工作。导师严谨求实的治学态度，朴实豁达的人生态度、孜孜不倦的敬业精神、循循善诱的教学风格是我受益终身。在本攻读研究生期间，不仅丰富了我的知识面，拓展了我的视野，并将这些知识运用到课题研究中，更是提高了我的实际工作能力。在此我衷心地感谢各院老师，认真负责的教学使我受益匪浅。

在研究课题过程中得到了师姐肖美华，师弟梁金泉，师妹高敏、马永卓、许虎，聂云峰老师和刘琳岚老师的大力帮助，也得到了实验室的硕士吴文俊，尹志军的帮助。在这里向他们表示深深的感谢。

最后，还有一份深深的感谢要特别的给我的父母，他们一如既往的关爱永远是我人生的支柱。

南昌航空大学硕士学位论文原创性声明

本人郑重声明：所呈交的硕士学位论文，是我个人在导师指导下，在南昌航空大学攻读硕士学位期间独立进行研究工作所取得的成果。尽我所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中作了明确地说明并表示了谢意。本声明的法律结果将完全由本人承担。

签名：_____日期：_____

南昌航空大学硕士学位论文使用授权书

本论文的研究成果归南昌航空大学所有，本论文的研究内容不得以其它单位的名义发表。本人完全了解南昌航空大学关于保存、使用学位论文的规定，同意学校保留并向有关部门送交论文的复印件和电子版本，允许论文被查阅和借阅。本人授权南昌航空大学，可以采用影印、缩印或其他复制手段保存论文，可以公布论文的全部或部分内容。同时授权中国科学技术信息研究所将本学位论文收录到《中国学位论文全文数据库》，并通过网络向社会公众提供信息服务。

（保密的学位论文在解密后适用本授权书）

签名：_____导师签名：_____日期：_____