

分类号_____

学校代码 10487

学号 M200972518

密级_____

华中科技大学

硕士学位论文

基于遗传和蚁群的自适应

路径规划算法研究

学位申请人: 丁寅

学 科 专 业: 计算机应用技术

指 导 教 师: 胡迎松 副教授

答 辩 日 期: 2012 年 1 月 15 日

**A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering**

**Research on Self-adaptive Path Planning
Algorithm Based on Genetic and
Ant Colony Algorithms**

Candidate : Ding Yin

Major : Computer Application Technology

Supervisor : Associate Prof. Hu Yingsong

Huazhong University of Science and Technology

Wuhan, Hubei 430074, P. R. China

January, 2012

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到，本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本论文属于 ☐ 保密，在_____年解密后适用本授权书。
☐ 不保密。

（请在以上方框内打“√”）

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

华中科技大学硕士学位论文

摘要

智能寻径问题源自几何学领域。当前随着机器人寻路、道路导航、虚拟现实、计算机游戏等领域研究逐步深入，该问题成为了一个很重要的研究课题。人工智能路径规划方法因其具有全局搜索、信息反馈和自我学习等优点，对它们的研究显得尤为重要。其中仿生学算法是人们受到生物进化和生物日常行为启发研究出的用来解决复杂优化问题的方法。其中包括遗传算法，神经网络，蚁群算法等知名算法。

介绍了当前几种常用的路径规划算法，并比较了它们的优势和不足。重点介绍了人工智能算法中的遗传算法和蚁群算法。遗传算法具有很强的快速全局搜索能力、鲁棒性，但是无法利用系统的反馈信息，容易在后期做大量无谓的冗余迭代，导致收敛速度下降。蚁群算法具有很好的信息反馈性，但是由于初期信息素匮乏导致求解速度较慢。给出了一种基于两种算法自适应动态融合的方法，前期利用遗传算法生成一组解作为蚁群算法的初始信息素分布，后期用收敛速度参数使蚁群算法自适应地融合遗传算子。不仅提高了收敛速度，而且改善了蚁群算法后期收敛过快导致陷入局部最优的问题，增强了整个算法的全局搜索性。

利用栅格法构建路径规划环境，在多种障碍物环境下进行仿真实验。通过大量实验对自适应动态融合算法的各个阶段参数对算法的影响以及它们的协同关系进行研究，得出了较为优化的参数组合方法。同时，对比基本融合算法和给出的自适应动态融合算法，在有效性、可行性以及算法性能方面得出了较为满意的结果。

关键词：路径规划，遗传算法，蚁群算法，自适应，融合

华中科技大学硕士学位论文

Abstract

Intelligent path finding problem stems from the field of geometry. With the gradual in-depth study in the areas of robot path finding, road navigation, virtual reality and computer game, intelligent path finding problem comes to an important research issue. The research on artificial intelligence methods is particularly important because of its global search, information feedback and self-learning etc. Bionics algorithms are that people gain inspiration from biological evolution and daily behavior of biological to solve complicated problems. Including genetic algorithms, neural networks, ant colony algorithm and other well-known algorithms.

First, describes several commonly used current path planning algorithm, and compare their advantages and disadvantages. Focuses on artificial intelligent path planning algorithms, genetic algorithm and ant colony algorithm. Genetic algorithm has a strong rapid global search capability, robustness, but can not use the feedback information in the system, easy to do a lot of unnecessary redundant computing at the latter part of algorithm, leading to decreased convergence rate. Ant colony algorithm has a good ability of information feedback, but solve slow due to lack of initial pheromone. Presents a dynamic self-adaptive fusion algorithm based on two methods. It use genetic algorithm to generate a solution as the initial pheromone distribution in the early, then self-adaptive parameter fuses genetic operators dynamically in the latter part of ant colony algorithm. Not only improve the convergence rate of solution, and improved ant colony algorithm converges too quickly to lead into the global optimum situation, enhance the global search ability of entire algorithm.

Through building the path planning environment by grid method and doing simulation experiments under a variety of environment, obtained more optimized method of parameter combinations by studying on the influence and relationship of parameters in every stage of the self-adaptive fusion algorithm. At the same time, compared to the basic fusion algorithm and the proposed self-adaptive fusion algorithm, obtained a more satisfactory result in the aspect of algorithm performance and effectiveness.

Key words: Path Planning, Genetic Algorithm, Ant Colony, self-adaptive, Fusion

华中科技大学硕士学位论文

目 录

摘 要.....	I
Abstract.....	II
1 绪论	
1.1 课题背景	(1)
1.2 国内外研究现状	(2)
1.3 研究内容与组织结构	(5)
2 路径规划智能算法相关技术	
2.1 遗传算法基本理论	(7)
2.2 蚁群算法基本理论	(11)
2.3 基本融合算法框架	(14)
2.4 本章小结	(16)
3 基于遗传和蚁群的自适应路径规划算法的设计	
3.1 自适应融合算法概述	(17)
3.2 遗传算法阶段	(18)
3.3 蚁群算法阶段	(23)
3.4 自适应动态融合阶段	(25)
3.5 算法总体流程及框架	(28)
3.6 本章小结	(30)

华中科技大学硕士学位论文

4 算法实现及性能测试	
4.1 算法实验平台	(31)
4.2 算法参数组合测试	(32)
4.3 算法性能测试	(39)
4.4 本章小结	(43)
5 总结与展望	
5.1 本文主要工作	(44)
5.2 进一步研究方向	(45)
致 谢.....	(46)
参考文献.....	(47)
附 录 攻读学位期间发表论文目录	(51)

华中科技大学硕士学位论文

1 绪论

路径规划是智能机器人、道路导航、虚拟现实领域里的一个重要问题。研究路径规划技术以及设计高效的路径规划算法，对增强机器人在复杂和危险环境中操作的有效性，路径诱导的准确性和及时性以及虚拟角色寻路的真实性有重要意义。本章首先介绍了课题的研究背景，然后分析了国内外研究现状，最后对本文的主要研究内容和文章的组织结构进行了说明。

1.1 课题背景

智能路径规划问题起源于计算几何学。在 70 年代中期，随着对智能机器人、道路导航、虚拟现实以及计算机游戏的研究的深入，各个领域中的智能寻路逐渐成为了一个很重要的课题。在不同领域的应用中，智能路径规划一般分为智能机器人寻路和虚拟现实两大类。随着当前研究的深入，路径规划算法已经由原来简单的求最短路径演化到了现在需要考虑到各种情况的智能寻路。对于寻路本身必须到考虑地形的复杂性，障碍的多样性，以及寻路主体的本身特性。除此之外，通常还会需要考虑到寻路算法的时间复杂度，空间复杂度，搜索结果的全局性和局部性，搜索分支的平衡性以及寻路空间的构建方法对算法的影响等等。这些都使得智能的寻路成为了一个很具有挑战性的课题，它通常会需要将人工智能、计算机图形学，等各个学科的知识综合运用，才能解决问题。

现阶段，三维渲染的硬件设备和图形质量已发展到一定程度，各个领域 AI 的应用已经成为决定一个系统和一款软件的重要的评价因素。对于复杂的场景，路径搜索是整个体系中较慢的一个步骤，因此成为对系统进行性能优化所关注的重点之一。各种场景下的路径搜索是智能机器人角色所应该具备的最基本的能力。同时，路径搜索也是一项非常具有挑战性的任务，这来源于场景的复杂性、建模环境单元格数量的巨大性、对真实感的要求，以及计算机计算资源和内存资源的限制。智能机器人。路径搜索成为了程序员一道必须逾越的关卡，在虚拟现实领域也存在同样的问题。

华中科技大学硕士学位论文

1.2 国内外研究现状

路径规划方法首先被提出是源于 70 年代中期的机器人路径规划问题。经过多年的研究和改进,很多方法也被广泛用于虚拟对象的寻路问题中。长久以来,国内外学者和研究人员在该领域做了许多工作,获得了较好的解决效果,其中有一些研究成果具有重大意义。在这些成果中,所研究的路径规划算法种类繁多,并且针对不同的应用,算法各有侧重点,这些都很好的推动了智能寻路问题的发展。

传统方法有:栅格法,单元分解法,可见点法,自由空间法,人工势场法,A*算法,D*算法等方法。

栅格法:目前采用最多的一种方法,它是将寻径空间以栅格进行划分。栅格是一种具有二值信息的单元,有时会采用四叉树或者八叉树的结构^[1]。环境内的一切障碍物和路径都以栅格表示,所以路径规划就变成起点和终点两个栅格点之间的最短路径问题。该方法具有表达简洁、规范、具有一致性、简单容易实现等优点。栅格结构适合计算机的存储方式,信息更新快,在简单环境下该方法简便快捷。但是随着场景的复杂和对分辨率要求的提高,其存储空间和计算时间都以指数级增长,算法效率会急剧下滑^[2]。

单元分解法:对寻径空间进行分层表示的一种结构方法,如采用 4 叉树、8 叉树对局部区域进行更进一步的划分,达到精确寻路目的^[3]。该方法在环境中不同区域采用不同分辨率进行划分,在某种程度上降低了存储空间。但当障碍物较大时,树结构往往很大,叶子节点数和障碍物边长成正比,大量消耗内存且不易于管理^[4]。

可见点法:又名可视图法。该方法中把寻径对象视为一个点,然后把路径起点,终点,寻径对象以及多边形障碍物的顶点进行连接,要求所有连接是可通的。求出起点到终点的最短路径^[5]。该方法对路径点的设置有很大依赖作用,决定了算法的性能,当障碍物较多时各点连线复杂交错,所以不能保证在任何时候都找到一条最优路径,且当起点和终点发生改变后需要重新构造可视图。另外在该方法中,寻径对象被看做是一个点,忽略了其自身半径,在移动过程中很可能发生碰撞,降低安全性和可靠性^[6]。

自由空间法:采用一系列定义好的基本图形,如广义锥形、凸多边形,对寻径空间进行划分,要求基本图形能覆盖整个障碍物以免发生碰撞。划分完毕后将空间构建成连通图,然后通过对图的搜索来规划路径。该方法比较灵活,划分得当往往会

华中科技大学硕士学位论文

简化计算，且改变起点和终点不用重新构造连通图。缺点是随着障碍物的增多不是任何时候都能得到最短路径^[7]。

人工势场法：结合力学领域的一种方法，由 Khatib 提出了一种虚拟力的概念。该方法在寻径环境中建立虚拟的人工势场，而寻路对象就在势场环境中受到虚拟力来指引其运动。其基本思想是目标点产生吸引力，障碍物产生排斥力，距离越近势场力越强，虚拟角色在合力的指引下确定运动方向和加速度^[8]。人工势场法数学公式简明、易分析，算法的时间复杂度低，所以具有很高的采用率。但是该方法容易选入局部最优，全局搜索性不强。且在障碍物前会产生震动和在局部小区域产生来回摆动的现象。

A*算法：由 HART 等人于 1968 年提出的，主要用于静态路径规划，是目前国内外公认的静态寻路中最优的算法^[9]。在计算机游戏和虚拟现实领域运用比较广泛。该算法是在 Dijkstra 算法的基础上加上启发式搜索。其中引入了启发式函数 $h(x)$ ，综合考虑当前代价和估计代价的启发式代价来选择下一个移动的点^[10]。

D*算法：动态 A*算法（Dynamic A Star），由卡内及梅隆机器人中心的 Stentz 在 1994 和 1995 年两篇文章提出。该方法是先用 Dijkstra 算法规划出一条路径，然后每移动一步就检测是否有障碍物，如果存在则用 A*算法进行重新计算否则继续沿着原路前进^[11]。该方法在动态环境中具有比较好的效果，但当场景大而复杂的情况下，该算法往往显得力不从心^[12]。

随着近年来人工智能领域的快速发展。人工智能方法应为其各自拥有的优势，逐渐被各界所重视，利用到路径规划的研究中。常用的智能路径规划方法有：遗传算法，神经网络算法，模糊逻辑控制算法，蚁群算法和粒子群算法等等。

遗传算法：遗传算法（Genetic Algorithm, GA）最初由美国的 Michigan 大学的 J.Holland 教授于 1975 年首先提出来的，是模拟自然选择和遗传学机理的生物进化过程的计算方法^[13]。它通过模拟自然进化过程搜索最优解。该方法对于求解各种优化问题提出了一种通用的框架。首先对目标问题进行编码为染色体，确定适应度函数。然后选择合适的遗传算子，通过选择、交叉、遗传等操作进行数代进化得到较优的结果^[14]。该算法的特点是直接对结构对象进行操作，没有求导和函数连续性的要求；具有快速的全局寻优搜索能力以及鲁棒性；采用概率化的优化方法，算法能自动获取和指导搜索空间，通过适应度函数自适应地调整搜索方向。但是该算法在求解后期往往会做大量冗余无用操作，降低算法效率^[15]。

华中科技大学硕士学位论文

对于障碍物环境中的寻路问题,已有的算法中,有些在实时性方面具有优势,但对于全局最优解的寻找却能力不足。该问题实际上可以转化为一类带约束的优化问题。因为遗传计算寻优的全局性和高效性,各界学者陆续展开了用遗传算法进行路径规划的研究,并取得了一定的成果^[16]。并通过大量的数字实验对于遗传算法解决2维障碍规避问题的各个参数的选取和算法停止世纪都做了有益的探索,验证了遗传算法在平面障碍环境中寻路是有效的^[17]。

模糊逻辑控制算法:是一种模拟驾驶员驾驶时的行为研究得出的方法。他将生理学中的从感知到动作的过程融入了模糊控制之中。使该算法在具有鲁棒性的基础上具有人类专家的行为。它适用于实时变化的未知环境,不需要完整的环境建模,对于通常用定量的方法来解决复杂的问题或是当外界只能提供定性近似的、不确定信息数据时非常有效^[18]。正因为模糊逻辑往往根据人们的日常经验预先制定的,所以具有灵活性差和无法自学习等缺点,但是算法的速度较为乐观。

神经网络算法:神经网络(Artificial Neural NetWork, ANN)算法是一种模仿生物神经活动的算法,其最大的特点是具有自适应学习能力和鲁棒性,同时它是一个高度并行的分布式系统。Ashraf A.Kassim 等人于1992年提出了基于音波扩张的神经网络结构用于路径规划^[19]。该文提出了一种一般的神经网络结构,专门设计用于实现对于路径规划有用的潜在操作,该神经网络把需要导航的区域作为输入,然后产生人工潜在的区域用于寻路。2003年 Dmitry V. Lebedev, Jochen J. Steil 等人提出了一种动态音波扩张神经网络^[20]。该方法产生潜在有用的动态距离用于时间变化的环境。该方法最大的特点就是可以适应环境的变化。

蚁群算法:蚁群算法(Ant Colony Optimization, ACO)由 Marco Dorigo 于1992提出,其灵感来源于蚂蚁在寻找食物过程中发现路径的行为。它模拟蚂蚁群在巢穴和目标物之间进行寻路,每只蚂蚁都是相互独立的,但是它们走过路径时会释放一定量的信息素,同时这些信息素会随时间挥发,释放信息素的量与蚂蚁距目标的距离成反比。每只蚂蚁寻路时会倾向于信息素多的方向。随着时间的推移,越来越多的蚂蚁会迅速收敛与一条路径。蚁群算法结合了分布式计算、正反馈性、启发式贪婪搜索等特点,成为了一种新型的求解优化问题的通用方法^[21]。实际上该算法也是一种进化算法,并且具有更多的优良性质。

T Stutzle, H Hoos 提出了最大最小蚂蚁算法,该算法是目前为止解决 TSP, QAP 等问题最好的蚁群优化算法^[22]。国内同时也有大量相关蚁群算法的研究,吴斌,史忠

华中科技大学硕士学位论文

植提出了一种分段求解的方法,取得了良好的效果^[23]。陈峻,沈洁,秦玲则提出了一种均匀分布的自适应模型^[24]。

粒子群优化算法:粒子群优化算法(Particle Swarm Optimization, PSO)由 Eberhart 和 Kennedy 于 1995 年提出的一种从鸟类捕食行为中得到启示的进化计算技术。它也是一种基于群体的优化算法。相较 GA 更为简单,因为它无需算子操作,而是通过计算群中每个粒子的适应值来确定解空间中的最优粒子。然后其他粒子们通过两个“极值”不断更新自己,从而达到逼近最优粒子的目的^[25]。但是它已经被证明不是一个全局收敛的算法。为了解决这个问题,后来产生了量子群算法,它在粒子群算法的基础上做了改进,在全局收敛性上大大优于前者。

1.3 研究内容与组织结构

本文致力于研究智能路径规划算法,总结了国内外学者关于碰撞检测算法的研究工作,本文提出了一种基于遗传算法和蚁群算法的自适应动态融合的方法,不仅提高了了解的收敛速度,而且改善了后期蚁群算法收敛过快导致陷入局部最优的情况,增强了整个算法的全局搜索性。本文主要的研究内容如下:

(1) 对智能路径规划方法进行深入探讨,主要介绍了遗传算法和蚁群算法的基本概念、设计方法和基本流程。

(2) 提出了一种基于遗传和蚁群的自适应路径规划算法。该算法在汲取遗传算法和蚁群算法不同阶段收敛速度优势的基础上加入了收敛速度参数,使得整个算法后期可以自适应的融合遗传算子。在牺牲很少计算时间的基础上大大降低了算法陷入局部最优的概率,提高了算法的稳定性。

(3) 本文在栅格法建模的基础上,设计并实现了本文提出的路径规划算法。并在多种障碍物环境的情况下对该算法的可行性和有效性进行了仿真实验。同时研究了自适应融合算法各个阶段参数大小和协同关系的对算法的影响,确定了参数的最优组合范围。最后在相同参数的基础上同基本融合算法做了比较。

本文的组织结构如下:

第一章是绪论部分,首先介绍了课题的研究背景,然后对当前国内外研究现状及存在问题进行了详细地论述,最后阐述课题的主要研究内容和论文组织结构。

第二章对智能路径规划方法进行深入探讨,主要介绍了遗传算法和蚁群算法的发展,分析了它们的基本原理和算法步骤。最后介绍了基本融合算法的思想。

华中科技大学硕士学位论文

第三章详细地给出了基于遗传算法和蚁群算法的自适应融合路径规划方法的设计思路和算法模型，在结合了两种方法优点的基础上，引入了算法收敛速度的参数，使得算法后期可以自适应的融入遗传算子操作，大大降低了算法陷入局部最优的概率。

第四章是系统实现及测试部分，介绍了算法实现的平台以及测试算法的寻路空间的构建，通过大量实验研究了本文算法各个阶段参数大小和协同关系的对算法的影响。最后在参数相同的基础上同基本融合算法做了比较，证明本文提出算法的优越性。

第五章是全文的总结部分，阐述本论文的主要工作，并在其基础上对下一步的工作提出展望。

华中科技大学硕士学位论文

2 路径规划智能算法相关技术

目前, 智能路径规划已经成为智能机器人、道路导航、虚拟现实以及计算机游戏等重要领域中不可避免的问题。在人工智能方法中遗传算法和蚁群算法因其运用的广泛性成为研究最多两种方法。本章首先介绍了 GA 和 ACO 的基本原理、设计方法、算法框架; 然后介绍了当前基于两种算法的一种基本融合算法。

2.1 遗传算法基本理论

生物系统可以通过自身的不断演化来完成许多人类看似复杂的问题。遗传算法就是模拟生物在自然界优胜劣汰的自然规律的一种全局概率搜索方法。它从初始解开始通过不断的迭代计算优化当前解, 知道得到令人满意的结果^[26]。

2.1.1 遗传算法概述

遗传算法最早由美国密西根大学的 John.H.Holland 教授提出。依据达尔文进化论“适者生存”的理论, 模仿大自然进化过程中具有的全局性, 随机性和鲁棒性。它把所求的问题解表示为染色体, 染色体上存在基因。然后选择合适的适应度函数来估量染色体的优劣性。将自然选择、杂交、突变等遗传进化行为模仿为选择、交叉和变异等遗传算子, 利用算子对染色体种群直接进行操作。从而达到逐步寻优求解的目的^[27]。

与其他优化算法直接以关键变量和实际值作为操作对象不同, 遗传算法以染色体作为操作对象且直接以适应度作为搜索指导信息, 对求导和函数连续性没有要求。算法使用多个点的搜索信息, 具有快速的全局寻优搜索能力以及鲁棒性。采用概率化的优化方法, 自动获取搜索空间, 通过适应度函数自适应地调整搜索方向。所以遗传算法提供了一套求解复杂系统问题的通用模型, 且不依赖于问题的具体领域, 所以具有较为广泛的科学应用^[28]。主要应用包括: 函数优化、生产调度、机器人学、图像处理等^[29]。

遗传算法是一种模仿生物进化过程的全局性随机方法, 所以算法中用到很多进化论和遗传学的概念, 其含义解释如下:

个体: 遗传算法优化的基本对象。

华中科技大学硕士学位论文

染色体：个体的编码。来表现出不同个体间的特性。

基因：组成染色体的元素，也就是染色体的各个编码，表示染色体的特征。

基因位置：基因在染色体的整个编码中所占的位置。

编码：表现型到基因型的映射。

解码：基因型到表现型的映射。

群体：若干个体的集合。

群体规模：群体中个体的数量。

适应值：适应度函数计算出来的数值，表示个体对环境的适应能力，此值越高表示该个体越优秀。

选择：选出此代中较为优秀的个体。

交叉：交换不同染色体上的基因。

变异：改变某个染色体上的基因。

进化：通过选择、交叉、变异等遗传算子对群体做一次操作，产生下一代新群体的过程。它们也反映了物种的进化、淘汰过程。

2.1.2 遗传算法设计方法

(1) 编码是将目标问题解映射到个体的染色体的编码空间中，是在设计遗传算法时首先遇到的问题。编码方式决定了染色体个体的排列方式的同时，同样决定了个体空间的复杂度以及遗传算子操作的实现与效率。当前最常采用的评估编码策略有以下三个要求：

- ① 完备性：目标问题空间中的所有解都可以由染色体空间编码所映射。
- ② 健全性：算法空间的染色体编码能对应所有目标问题解空间中的候选解。
- ③ 非冗余性：染色体和候选解一一对应。

迄今为止主要由三大类编码方法：浮点数编码、二进制编码、符号编码。其中二进制编码因为其简单性是 GA 中最常用的编码。但是它可能会因为某个基因的变异操作造成与原有染色体较大的差异，降低了 GA 的局部搜索性，且当搜索空间较大时，染色体编码长度也会过大导致整个算法效率下降。对于搜索空间较大，求解精度要求比较高的问题可以采用浮点编码方法。而符号编码的优势在于自身具有的背景知识。

(2) 适应度函数是评价个体优劣的依据，也是该个体繁殖后代的能力。同时又

华中科技大学硕士学位论文

是算法运行导向和算法复杂度的重要指标。遗传算法在搜索寻优过程中不需要其他外部信息，只用评估函数来衡量个体的优劣，并作为本轮遗传操作的唯一依据。所以根据不同的目标问题设计一个合适的适应度函数非常重要。

在 GA 前期少数极优秀个体可能被遗传下来，导致后期充斥着整个种群。使得种群的多样性降低。种群内的交叉算子操作失去意义，所以适应度函数的设计前期应该能适当降低优势个体和劣势个体之间的差异，避免算法早熟收敛于局部最优解。而算法后期种群内个体的相似度增大，个体之间趋于近似，个体相互竞争力下降。选择算子退化为随机选择，不利于精英个体的遗传到下一代。所以适应度函数的设计的设计在算法后期应该提高个体之间的差异从而提高种群内的竞争力。比较常用的三种适应度函数方式如下，其中 F 为适应度函数， f 为目标函数：

- ① 线性变换： $F = af + b$, a, b 为实数
- ② 乘幂变换： $F = af^k + b$, k 为幂指数
- ③ 指数变换： $F = \alpha \exp(\beta f)$, α, β 为正实数

(3) 遗传算子是 GA 的主体，决定了其全局随机搜索的能力。一般包选择，交叉和变异三个基本操作。其中选择算子保证了优势个体基因能够有几率得到保存，交叉算子是产生新个体的主要方法，具有全局搜索能力。变异算子在局部范围内进行搜索可作为辅助算子。遗传算法通过交叉和变异两种算子相互配合达到搜索效果的最优化。

选择算子设计：又名复制算子。该算子遵循达尔文进化论里的优胜劣汰的原则，使适应度高的个体被选择进入下一代的概率大，保证了精英个体的基因得到保存，通过该方法使群体的适应度不断提高，最终达到要求的一组解。选择算子设计不当可能导致无法保留精英个体到下一代，致使整个群体停止进化，或者前期过分加强精英个体，使得后期这些个体充斥着整个种群，降低了种群的多样性使算法收敛于局部最优。比较常见的选择算法有轮盘赌法、锦标赛法、排序选择法等。

交叉算子：以一定的概率随机的选择一对个体，以某种的方法交换一部分染色体。从而形成新的个体，这些下一次中往往就有适应度更高的个体。交叉算子是遗传算法中最为重要的算子，因为其全局搜索能力作为主要算子，同时也是产生新个体的主要方法，保持了种群的多样性。主要的交叉策略有：多点交叉、单点交叉、二进制交叉、均匀交叉等。其中以单点交叉和多点交叉最为常见。

变异算子：以一定的概率，选取个体的染色体上某些基因进行变化产生新的个体。

华中科技大学硕士学位论文

变异算子也是产生新个体的方法，与交叉算子相比它是在一定范围内进行基因替代的随机方法，是一种辅助产生新个体的方法，决定了算法的局部搜索能力。与交叉算子配合取得较好的效果，使得算法在全局搜索和局部搜索达到平衡。

(4) 算法终止的条件是算法达到之前定下的标准而停止，将最近的一代种群作为全局近似最优解输出，当前使用的两种方法为：

①遗传算法迭代次数也就是进化种群代数达到预先设定的值，可以停止算法。

②当前种群的平均适应度达到预先制定的标准后，我们也认为当前种群中的个体为我们需要的满意解，可以停止算法。

③指定代数的种群进化率连续小于某个预定的值后，说明遗传算法进入到后期冗余迭代做功甚微的阶段，可以停止算法。

遗传算法中各个参数的设计和关联较为复杂，而且参数的设定直接影响到算法性能和准确性^[30]，所以在设计过程中对参数的研究是一个重要课题。

2.1.3 遗传算法基本流程

算法的基本流程主要包括：染色体编码、生成初始种群、选择、交叉、变异等遗传算子操作。基本步骤如下：

- (1) 设计编解码方法，把目标问题从解空间映射到染色体的编码空间中；
- (2) 设计确定适应度函数；
- (3) 设置遗传算法的各种参数值，包括：初始种群大小，各种遗传算子发生的概率。同时确定遗传算子的具体操作方法；
- (4) 初始化第一代种群；
- (5) 对种群中的个体进行适应值计算；
- (6) 根据适应值对种群做选择、交叉、变异等遗传算子操作。进而形成下一代种群；
- (7) 判断算法是否符合终止条件，如果是则结束算法。否则跳到(5)继续执行。

图 2.1 为遗传算法的一般流程图。

华中科技大学硕士学位论文

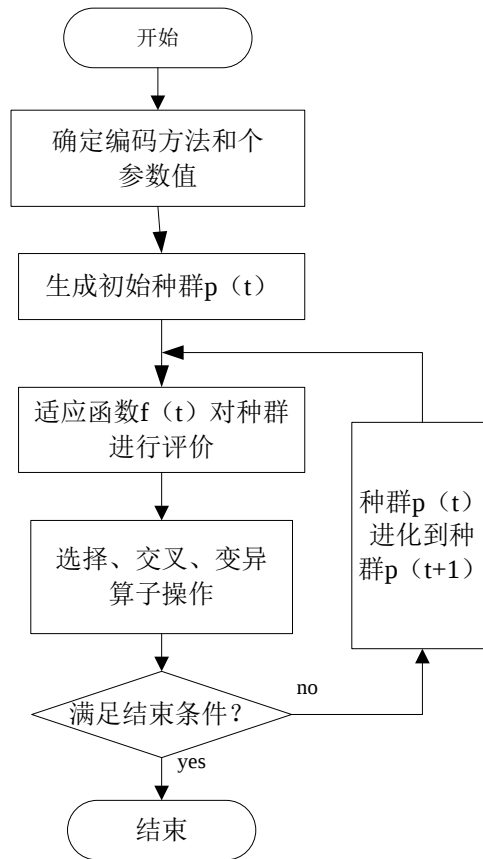


图 2.1 遗传算法流程图

2.2 蚁群算法基本理论

蚁群算法同样是一种仿生学方法，它模仿蚂蚁在觅食过程中走过的路径会留下信息素的行为，把每只蚂蚁联系起来。该算法就是通过信息素来进行正反馈，最终使得整个蚁群趋于最优的路径上。该算法应用简单，具有较好的并行性和鲁棒性，用于工程设计、组合优化多个领域^[31]。

2.2.1 蚁群算法概述

ACO 算法模仿蚂蚁群在觅食的过程，在寻找食物时每只蚂蚁都是相互独立的，但是它们走过的路径时会释放一定量的信息素，同时这些信息素会随时间挥发，释放信息素的量与蚂蚁距目标的距离成反比。每只蚂蚁寻路时会倾向于信息素多的方向进行移动，但是还有些蚂蚁不遵守规则，它们会另辟蹊径，这些蚂蚁增加了算法的搜索范围和随机性，这样整个系统就通过信息素把整个蚁群联系在了一起。随着

华中科技大学硕士学位论文

时间的推移，越来越多的蚂蚁会被信息素所吸引过来从而造成更多信息素的积累，如此正向反馈下去，算法将迅速收敛与一条路径。蚁群算法是一种求解组合最优化问题的新型通用启发式方法^[32]。ACO 算法有如下几个特点：

(1) 并行性：每只蚂蚁的运动都是独立的，相互之间没有依赖和影响。整个系统也没有依赖于某一个个体，蚂蚁之间只通过信息素进行交流。在没有统一指挥的情况下，蚂蚁之间相互协作构造可行解。

(2) 正反馈性：蚂蚁之所以聚集，是应为信息素的积累。越多的信息素会吸引更多蚂蚁，同时越多的蚂蚁又会得到更多的信息素，正反馈让初始信息素不断得到加强，引导着整个系统朝着最优解的方向进化^[33]。

(3) 自组织性：理解为系统在无外力的作用下从无序到有序的过程，蚁群算法初期每只蚂蚁都无序的进行移动觅食，后期经过算法迭代，每只蚂蚁都找到趋于最优解的路径，这个过程无外力作用完全是算法自组织完成的。

近几年已有学者探索性的将一群算法用于机器人领域的路径规划^[34]。包括单个机器人和多机器人协作规划^[35]。

2.2.2 蚁群算法设计方法

蚁群算法最先是在解决 NP-hard 经典问题中使用的。Dorigo 将它应用于旅行商问题(Traveling Salesman Problem, TSP)，取得了较好的效果，旅行商问题简单的说就是求一条遍历 n 个城市的最短闭合回路^[36]。

算法需要在内部添加禁忌表 $tabu_k$ ，用来记录每只蚂蚁走过的节点，以避免蚂蚁重复经过同一个节点。对于 m 只蚂蚁和 n 个城市，第 k ($k=1,2,\dots,m$) 只蚂蚁对下一步可能要经过的每个节点的概率计算公式为：

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{j \in allowed_k} [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta} & j \in allowed_k \\ 0 & j \notin allowed_k \end{cases} \quad (2.1)$$

式中 $allowed_k = \{C - tabu_k\}$ 为蚂蚁可以选择的城市节点，也就是它之前没有走过的节点。 α 为信息素启发式因子，反映了在选择移动方向时信息素对选择概率的影响，该参数越大，系统中各个蚂蚁之间的联系越紧密。 β 为期望启发因子，反应了选择路径长度对选择概率的影响，该参数过大选择就会倾向于当前最短路径的贪心算法。

华中科技大学硕士学位论文

$\eta_{ij}(t)$ 为启发式函数，表示 t 时刻 i 节点至 j 节点的期望，一般为路径长度的倒数：

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \quad (2.2)$$

当蚂蚁完成一次寻路后需要对系统中的信息素做调整，在 $t+n$ 时刻 i 到 j 路径上的信息素调整如下：

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2.3)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (2.4)$$

其中 $\rho(\rho \in (0,1))$ 表示挥发因子，也就是信息素的衰弱。 $\Delta\tau_{ij}(t)$ 表示该次迭代中路径 (i, j) 上的信息素增量， $\Delta\tau_{ij}^k(t)$ 表示第 k 只蚂蚁在该次迭代中留在路径 (i, j) 上的信息素。 $\Delta\tau_{ij}^k(t)$ 的求法因为信息素更新模型的不同而有所不同。基本分为：蚁量模型、蚁周模型和蚁密模型^[37]。

蚁量模型

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/d_{ij}, & i, j \in L_k \\ 0, & i, j \notin L_k \end{cases} \quad (2.5)$$

Q 是信息素浓度常数，该参数越大收敛速度越快，反之越慢。

蚁周模型

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L_k, & i, j \in L_k \\ 0, & i, j \notin L_k \end{cases} \quad (2.6)$$

L_k 为第 k 只蚂蚁本次寻得的路径总长度。

蚁密模型

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q, & i, j \in L_k \\ 0, & i, j \notin L_k \end{cases} \quad (2.7)$$

以上三种方法中，蚁周模型因为其利用整条路径长度来进行更新，具有更强的全局性，所以一般选取它作为信息素更新模型。

2.2.3 蚁群算法基本流程

蚁群算法一般流程基本步骤描述如下：

- (1) 设置蚂蚁数量 m ，最大循环次数 $N_{\max}$ ，初始化路径信息素量和禁忌表 $tabu_k$ ；

华中科技大学硕士学位论文

- (2) 将 m 只蚂蚁随机放在 n 个城市节点上;
- (3) 计算每只蚂蚁下一步可能移动到节点的概率;
- (4) 按概率选择移动方向并移动一步, 同时更新禁忌表;
- (5) 若蚂蚁遍历完所有城市执行 (6), 否则执行 (3);
- (6) 计算蚂蚁该次走过的路径长度, 更新全局信息素;
- (7) 若迭代次数达到 $N_{\max}$ 则退出算法, 否则调至 (2) 继续执行。

图 2.2 为蚁群算法的一般流程图。

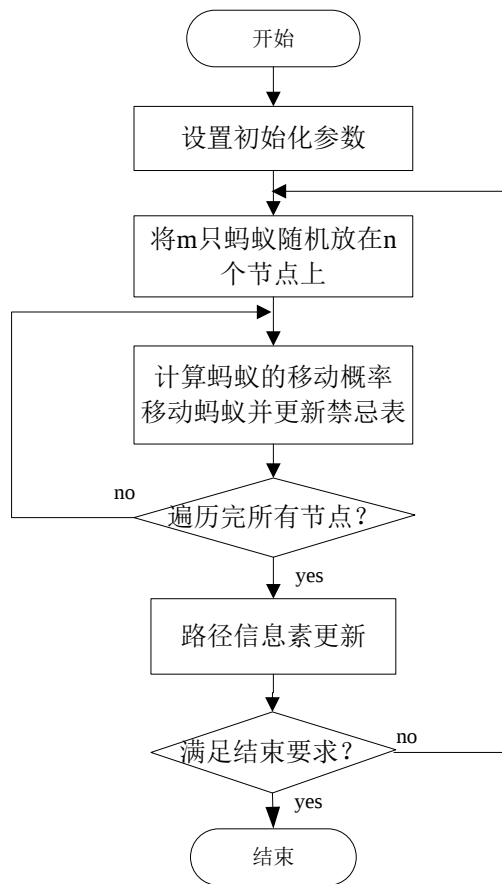


图 2.2 蚁群算法流程图

2.3 基本融合算法框架

之前两节介绍了遗传算法和蚁群算法这两种广泛应用的人工智能方法和它们各自的优势及特点。虽然两种算法都具有很多优点, 但是他们各自的缺点也是明显的, 导致了各自运行时占用的时间较长, 效率较低^[38]。

遗传算法的主要缺点:

华中科技大学硕士学位论文

- (1) 没有充分利用系统反馈信息，使搜索具有盲目性^[39]；
- (2) 算法求解到一定范围时往往做大量冗余迭代，向最优解收敛的速度大大降低，导致求最优解效率较低。

蚁群算法的主要缺点：在搜索初期信息素较为缺乏，导致搜索的初期信息素的积累时间较长，影响了效率^[40]。

通过对比遗传算法与蚁群算法的研究结果如图 2.3 所示，纵轴表示算法的收敛速度，横轴表示算法的运行时间。遗传算法在搜索初期具有较高趋于最优解速度的收敛速度，后期往往做大量无用迭代。而对于蚁群算法正好相反，它后的收敛速度快速的上升，而在之前由于前期信息素不足，导致搜索速度缓慢^[41]。

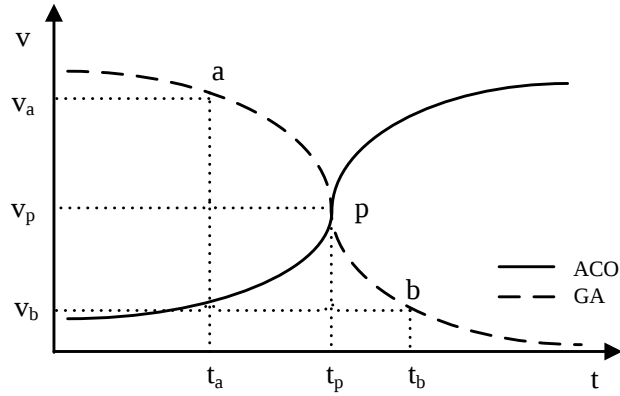


图 2.3 两种算法的时间速度曲线

所以，基本融合算法 (genetic algorithm ant algorithm, GAAA) 的基本思想就是^[42]：前期利用遗传算法的随机性和全局搜索性生成一组解作为蚁群算法初始信息素分布，也就是令两种算法在尽量靠近图 2.3 的 p 点进行融合。因为有了遗传算法提供的信息素积累，蚁群算法得以跳过前期信息素匮乏阶段，直接进入后期较为高效的阶段。该算法被验证在收敛速度上比单独的两种算法有明显的提高，基本融合算法框架图如下。

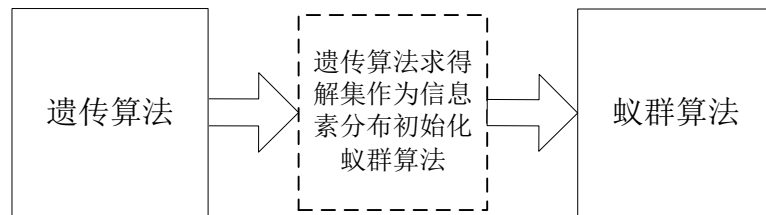


图 2.4 基本融合算法框图

华中科技大学硕士学位论文

该算法提出后被应用于多个研究领域，这些研究问题多事 NP-hard 问题。如旅行商问题^[43]，多协作任务分配问题^[44]等，都取得了良好的效果。

2.4 本章小结

本章介绍了智能路径规划算法的基本原理及其相关的一些技术。主要介绍了遗传算法和蚁群算法的基本原理，设计实现和算法流程，最后介绍了现阶段的基本融合算法的思想。这些技术将为下面两章中基于遗传和蚁群的自适应路径规划算法的设计和实现提供前提和基础。

3 基于遗传和蚁群的自适应路径规划算法的设计

根据 GA 遗传算法和 MMAS 最大最小蚂蚁算法各自的特点，有效的融合两种方法，提高了算法的收敛速度。且在基本融合算法的基础上对其进行改进，在算法后期的 MMAS 算法中引入收敛速度参数，自适应的融合入遗传算子操作，利用遗传算子的全局搜索性，在保证时间效率的基础上大大降低了后期蚁群算法由于收敛速度过快陷入局部最优的概率，使算法收敛与全局最优解。

3.1 自适应融合算法概述

现有的基本融合算法，其思想是汲取两种算法的优点，利用遗传算法前期全局搜索性强，收敛速度快的特点为蚁群算法生成初始的信息素分布。然后由蚁群算法在一定信息素导向的基础上进行正反馈寻径。在时间效率和求解效率都比较好的一种新的启发式方法。

但是上述融合算法在融合阶段两个算法是相互独立的，基于遗传和蚁群的自适应路径规划算法是在基本融合算法的后期自适应融入遗传算子，不仅在时间效率上优于单独的遗传算法和蚁群算法，而且改善了基本融合算法后期收敛过快导致早熟以及随机性和全局搜索性能低下的问题。该算法可以分为三个步骤：遗传算法生成初始路径信息素分布；蚁群算法寻路并按规则更新信息素；自适应融合遗传算子对路径进行更新优化策略。算法流程见图 3.1。

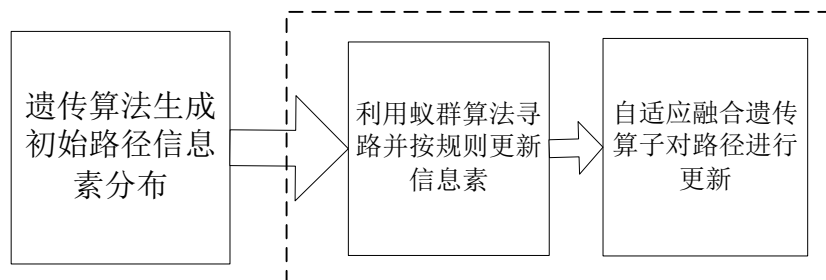


图 3.1 算法流程概述

(1) 遗传算法生成初始路径信息素分布：设寻路空间是二维的，本文采用最常用的栅格法对路径空间进行划分。然后确定路径编码方式以及适应度函数。生成初始种群后采用带有精英策略的遗传算法通过选择，交叉，变异遗传算子的操作对种

华中科技大学硕士学位论文

群进行遗传进化。待算法终止后得到若干满足条件的路径；

(2) 蚁群算法寻路并更新路径信息素：通过遗传算法得到蚁群算法中初始需要强化的路径后。为了体现对这些初始路径的加强，我们这里使用一个变种的最大最小蚂蚁算法（max-min ant system, MMAS）对路径进行初始化设定。然后通过该蚁群算法进行路径规划同时更新路径信息素；

(3) 自适应融合遗传算子对路径进行更新：基本融合算法虽然在收敛速度上有很大的改进，但是在后期蚁群算法往往会出现收敛速度过快的现象，使结果陷入局部最优，本文提出了一种基于收敛速度参数 γ 的自适应动态融合遗传算子的方法（adaptive genetic algorithm ant algorithm, AGAAA），保证了在蚁群算法收敛速度过快的情况下，利用遗传算子的随机性和全局搜索性对算法后期生成的路径进行一次遗传算子操作。如此往复，直到蚁群算法满足结束条件退出。

3.2 遗传算法阶段

本文采用最常用的栅格法对 2 维寻路空间进行划分。栅格最常用的表示方法有如下两种：坐标法和序号法。

(1) 坐标法：在上个上建立直角坐标系，以左上角为原点，水平向右是 X 轴正方向，竖直向下是 Y 轴正方向，任一栅格都可以用坐标(x,y)来表示， $M \times N$ 的坐标栅格划分方法见图 3.2。

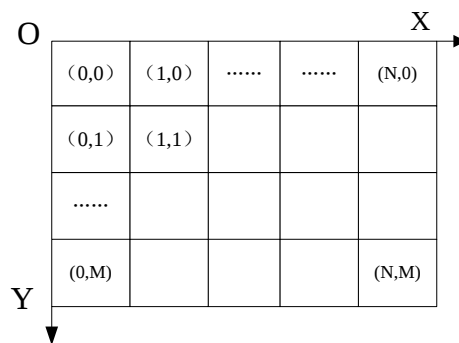


图 3.2 坐标栅格划分方法

(2) 序号法：把栅格按从左至右，从上至下的顺序，从 0 号开始依次递增编号。

华中科技大学硕士学位论文

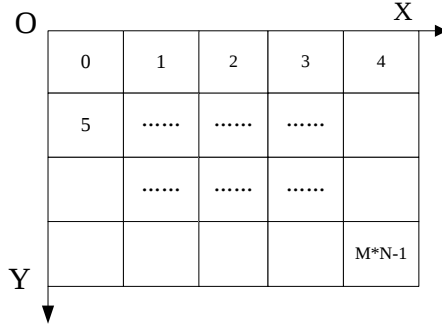


图 3.3 序号栅格划分方法

上述两种方法的映射关系为下式：

$$Num = x + N * y \quad (3.1)$$

$$\begin{cases} x = Num \% N \\ y = ROUND(Num / N) \end{cases} \quad (3.2)$$

其中%为去余，ROUND 表示取整。为了简化并加速遗传算子的计算，采用了坐标法对寻路环境进行构建。

3.2.1 生成遗传算法初始种群

为了简化路径编码方式，本文采用只记录每个栅格 Y 轴坐标的方法。即针对每个指定的横坐标，利用 rand 函数生成一个 $0 \sim (M-1)$ 的随机数来确定一个染色体基因，也就是说对于染色体的指定位置的基因的横坐标已经是确定的。同时每个染色体的开始基因和结束基因的编码是确定的，也就是起点和终点对应栅格的 Y 轴坐标。

针对此种编码方式，其种群中所有个体的位串长度均固定，即为栅格的列数 N。所以，在融合算法的前期遗传算法中采用是顺序的存储结构，不但节省了一半坐标点信息的存储空间，且可以快速的定位到染色体上指定位置的基因。例如在下面一个 4×5 的栅格中一个随机生成的染色体，染色体的存储结构和路径表示如图 3.4 和图 3.5 中的实线所示：

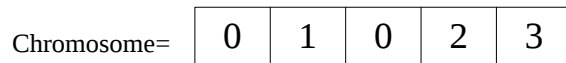


图 3.4 染色体的存储结构

华中科技大学硕士学位论文

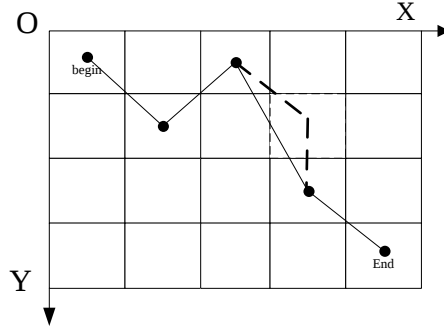


图 3.5 生成初始种群方法

对于初始种群的规模，其数量越大个体的多样性就越高，所以算法全局搜索性就好，同时陷入局部最优解的可能性就小。但是如果数量过大会导致算法的计算量急剧上升，影响系统运行的速度。一般情况下 $n=20\sim 200$ 。

3.2.2 适应度函数

适应度是遗传算法进化的重要指标。在复杂环境下对适应度函数的要求满足两个条件：即路径在约束范围内可通，且保证路径长度最短。本文提出了一种基于长度和阻碍程度相结合的适应度函数。如下式：

$$F = \frac{Q}{\alpha F_{length} + \beta F_{obstacle}} \quad (3.3)$$

其中 Q 为一常数， F_{length} 为路径的长度，这里包含可行栅格点和障碍栅格点的总和， $F_{obstacle}$ 为路径中包含的障碍栅格点的总和。我们规定寻径对象一次只能向所在栅格相邻的栅格移动一次，且斜向移动一次的距离为水平或竖直移动一次的 $\sqrt{2}$ 倍。可以推出 F_{length} 计算方法如式， $F_{obstacle}$ 的计算方法如式：

$$F_{length} = \sum_{i=1}^{N-1} f_{length}(i) \quad (3.4)$$

$$F_{obstacle} = \sum_{i=1}^{N-1} f_{obstacle}(i) \quad (3.5)$$

其中 f_{length} 以及 $f_{obstacle}$ 是针对每一段相邻 X 坐标的栅格之间的路径长度和路径所通过的阻碍栅格的个数。相邻 X 坐标的栅格点依照之前规定的一次移动一步的原则进行延展，使路径成为由连续的点所组成，以方便计算路径的适应度。延展策略见图 3.5 中虚线所示。可推得 f_{length} 的计算方法如下式：

华中科技大学硕士学位论文

$$f_{length}(i) = \begin{cases} 1 & , y_i = y_{i+1} \\ \sqrt{2} + |y_{i+1} - y_i| - 1 & , y_i \neq y_{i+1} \end{cases} \quad (3.6)$$

式(3.3)中的 α ， β 表示两种适应值得权重因子，其中 $\alpha + \beta = 1$ ， Q 为一常数，为了保证适应度值的范围。

本文提出的适应度函数没有一味的采用连通路程的值大于不通路径的值而是引入 $F_{obstacle}$ 的概念和路径长度一起进行评估。有如下两个原因：

(1) 是为了避免那些与障碍碰撞点很少但是路径越长度却很优秀的染色体被淘汰。如图 3.6 所示，路径 A 虽然不是连通路程，但是在作为蚁群算法信息素分布上，其指导作用明显优于路径 B。图中黑色表示障碍物。

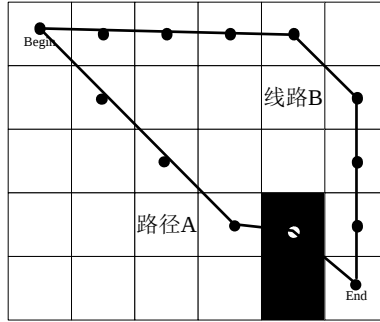


图 3.6 连通路程与非连通路程

(2) 遗传算法作为自适应融合算法前期部分，主要职责是提供蚁群算法初始信息素分布，并不需要保证路径的连通性，路径连通性可以再后期的蚁群算法进行保证。

3.2.3 遗传算子

(1) 选择算子：选择算子负责选择选择适应度较高的染色体，确定产生多少子代来用于后续的遗传算子操作。常见的选择算子有：轮盘赌法，锦标赛选择法，排序选择法等。为了保证算法的全局搜索性，本文采用锦标赛选择法和精英策略相结合的选择方法：

①锦标赛选择法：该方法随机的从父代中选择 N 个染色体，然后将适应度最高的选择出来，如此重复达到规定数量的子代即可。

②精英策略选择法：该策略是遗传算法收敛于最优解的一个基本保证。该方法的思想是当所有的遗传算子对当代种群进行完操作产生新一代种群以后，如果产生的新一代个体适应值小于之前的一代，就用若干父代最优个体替代相同数量的下一

华中科技大学硕士学位论文

算法的效率对路径不做连续性调整。路径连续性在蚁群算法中进行保证。

3.2.4 结束条件

为了保证融合的时机，本文采用预先设定迭代次数和进化率相结合的方法，设置遗传算法最大迭代次数 G_{Max} 和最小迭代次数 G_{Min} ，同时设置最大停滞代数 G_{Stop} 和最小进化率 G_{ratio} 。结束条件可描述为：

- (1) 进化次数达到 G_{Max}
- (2) 进化次数大于 G_{Min} ，且连续 G_{Stop} 的进化率都小于 G_{ratio}

两者满足一条则可以退出遗传算法阶段并进入蚁群算法阶段。

3.3 蚁群算法阶段

自适应融合算法中蚁群算法阶段的主要任务是接受遗传算法生成的一组解来初始化环境信息素分布。同时计算算法收敛速度自适应的融合遗传算子，从而求出趋近于全局最优路径解。

本文中提出的算法是在 MMAS 算法的基础上进行设计的。MMAS 算法和一般的蚁群算法相比主要有以下三点改进：

(1) 信息素限制：将整个环境的信息素量的范围限制在 $[\tau_{min}, \tau_{max}]$ 范围之内，当前信息素小于最小值时 $\tau = \tau_{min}$ ，同理当大于最大值时 $\tau = \tau_{max}$ ，可以避免信息素为 0 或者某条路径无限制积累造成局部最优的情况。

(2) 信息素更新：每次只对该次迭代最优路径或全局最优路径进行信息素加强。

(3) 信息素初始化：算法开始时将全局信息素都初始化为 τ_{max} ，这做可以增大搜索范围，避免算法早熟。

3.3.1 信息素初始设置

本文在基于 MMAS 算法的基础上进行适当修改实现蚁群算法与遗传算法的衔接。MMAS 算法中把各路径的信息素值设为最大 τ_{max} ，且某一路径点的信息素值只能在 $[\tau_{min}, \tau_{max}]$ 之间波动。当前信息素小于最小值时 $\tau = \tau_{min}$ ，同理当大于最大值时 $\tau = \tau_{max}$ 。这里我们设置初始信息素为

$$\tau = \tau_M + \tau_G \quad (3.7)$$

华中科技大学硕士学位论文

其中 τ_M 为蚁群算法的初始信息素我们这里规定 $\tau_M = \tau_{\min} + \frac{1}{2}(\tau_{\max} - \tau_{\min})$ ，而 τ_G 为常数表示遗传算法求解结果转换的信息素量。

3.3.2 移动概率计算模型

对于栅格法建模的寻径环境来说，蚂蚁在计算相邻可移动栅格的概率的时候，首先应该选择无障碍物的可行栅格，然后利用第二章中的式(2.1)计算每个可移动点的概率。对于栅格环境寻径，我们规定 $\eta_{ij}(t)$ 的值不同于 TSP 问题的简单的到下一点的路径，本文规定为其可行相邻点到终点的连线距离倒数，其中忽略障碍物的阻挡作用，简单理解为蚂蚁一厢情愿的朝着终点进行前进：

$$\eta_{ij}(t) = \frac{1}{\sqrt{(x_{\text{end}} - x_j)^2 + (y_{\text{end}} - y_j)^2}} \quad (3.8)$$

3.3.3 信息素更新模型

采用全局性最强的蚂蚁圈模型，即本次迭代中只有最短路径的蚂蚁才进行信息素更新，路径更新方程如下：

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (3.9)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (3.10)$$

式(3.8)中 $\rho(\rho \in (0,1))$ 表示挥发因子，则 $1-\rho$ 表示信息素的残留因子 $\Delta\tau_{ij}(t)$ 表示本次循环中路径 (i, j) 上的信息素增量， $\Delta\tau_{ij}^k(t)$ 表示第 k 只蚂蚁在本次循环中留在路径 (i, j) 上的信息素，其值由 Dorigo 提出的蚂蚁圈模型求，如式：

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L_k, & i, j \in L_k \\ 0, & i, j \notin L_k \end{cases} \quad (3.11)$$

其中 Q 为一个常数值，表示信息强度， L_k 表示第 k 中蚂蚁在本次循环中所走的路径长度。

3.3.4 陷阱策略和路径评价

蚁群算法在运行过程中蚂蚁容易陷入深度陷阱环境，从而导致它的僵死。所谓深度陷阱环境就是连续障碍物或连续障碍物与边界形成的只有一个出口的陷阱环

华中科技大学硕士学位论文

境，如图 3.8，对于此种环境必须做相应的处理，否则在复杂的环境中蚂蚁僵死的数量和概率将大幅度上升。

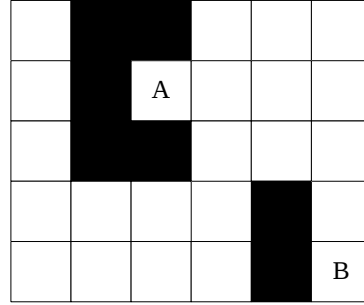


图 3.8 深度陷阱环境

对于此此种情况一般采用回溯法进行陷阱逃脱，当蚂蚁当前没有可行路径可走时，首先回退一步，再查找与当前所在栅格相邻的可行路径，如果有则继续寻径。否则再重复之前的回溯步骤，直到找到可行栅格位置。

因为在蚁群算法中，每只蚂蚁的行走路径是不可确定的，且蚂蚁寻得的路径一定是连通的，所以每条路径的长度有可能不同。而且为了方便路径的回溯以避免陷入深度陷阱，在蚁群算法中我们采用链表结构存储结构蚂蚁路径，在一个链表节点同同时保存栅格的 x 和 y 的坐标，以方便计算， m 只蚂蚁的路径存储结构如图 3.9

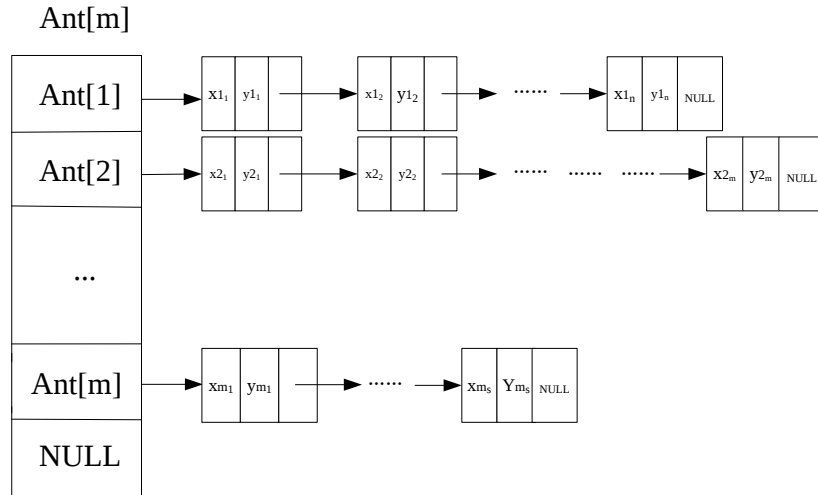


图 3.9 蚁群算法路径结构

3.4 自适应动态融合阶段

根据蚁群算法的定义，蚂蚁往往选择信息量较大的路径。当许多蚂蚁选中同一条路径时，该条路径就会被反复加强，从而蚂蚁的迅速聚集，造成阻塞和停滞现象，

华中科技大学硕士学位论文

表现为蚁群算法的早熟和陷入局部最优。

3.4.1 收敛速度参数

简单的遗传和蚁群融合算法虽然在收敛速度上有很程度的提高，但是后期蚁群算法容易由于收敛速度过快导致早熟，最终陷入局部最优解。针对这种情况，本文提出了一种基于蚁群算法收敛速度参数 γ 的自适应融合算法。该收敛速度参数即以蚂蚁的路径长度来表示每一轮算法优化的比率。 γ 的计算方法如下所示：

$$\gamma = \frac{\alpha\gamma_S + \beta\gamma_B}{\alpha\gamma_{SP} + \beta\gamma_{BP}} \quad (3.12)$$

式中 m 为蚂蚁的个数， γ_S 为每只蚂蚁在两次循环中求得路径的进化度之和， γ_B 为两次循环中最优蚂蚁的进化度的 m 倍。 γ_{SP} 为前一轮中所有蚂蚁的路径之和， γ_{BP} 为前一轮中最佳蚂蚁的路径的 m 倍。同理， α ， β 表示两个值的权重因子，其中 $\alpha + \beta = 1$ 。 γ_S 、 γ_B 、 γ_{SP} 、 γ_{BP} 分别由以下式子求得。

$$\gamma_S = \sum_{i=1}^m L_i^{n-1} - L_i^n \quad (3.13)$$

$$\gamma_B = m \sum_{i=1}^m L_{best}^{n-1} - L_{best}^n \quad (3.14)$$

$$\gamma_{SP} = \sum_{i=1}^m L_i^{n-1} \quad (3.15)$$

$$\gamma_{BP} = m \sum_{i=1}^m L_{best}^{n-1} \quad (3.16)$$

这里我们简单的利用路径长度的差来衡量进化程度。其中 L_i^n 为第 i 只蚂蚁在第 n 次循环求得的路径长度，而 L_{best}^n 为第 n 次循环中的最佳路径长度。我们称 γ_S 为平均进化度，而 γ_B 为最优进化度。

从式(3.13)，(3.14)可以看出在每次循环中每只蚂蚁求得路径的进化程度可能为正，也可能为负。为正说明路径进化，为负表示路径退化。所以所有蚂蚁的进化度之和可以全面表示种群的进化情况，从而为我们对算法收敛速度的判断提供有利的依据。

华中科技大学硕士学位论文

同时本文定义一个收敛速度阈值常数 v ，在每次循环中当参数 $\gamma > v$ 时则算法进入遗传算子部分，这里我们只使用交叉算子和变异算子进行操作。为了保证算法的效率，同时为了避免过多的遗传算子操作会造成过大的随机性和破坏原有的最有路径结构，该阈值同时控制着遗传算子操作的次数。

3.4.2 遗传算子

为了降低算法陷入局部最优的概率，同时有机的与 MMAS 相结合，在融合阶段的遗传算子只选用交叉和变异两种算子操作。

(1) 交叉算子：仍然选用单点交叉。但是与 GA 中的交叉不同，为了保证不破坏蚂蚁的路径结构，在交叉前要做以下处理。

①扫描两条随机选出的路径序列，检查它们是否有交点，如果有则跳至②，否则将两条路径重新放回待选池等待下一次随机选取操作；

②对交叉点以后的路径序列进行交换完成交叉操作，产生新的路径个体。

(2) 变异算子：交叉算子保证了路径的全局最优性，变异操作只对路径做优化和修饰作用，本文规定每个路径点的变异范围为它相邻的某个点，且变异后的点与它的前一个和后一个点做连续性操作，这样就保持了路径的连通性，也不需要复杂的后期处理计算。

3.4.3 融合模块描述

动态融合部分大体算法描述如下：

(1) MMAS 算法寻径并且 m 只蚂蚁都获得一条可行路径；

(2) 按式(3.11)计算该轮蚁群算法收敛速度参数 γ ，如果 $\gamma > v$ 跳至 (3)，否则调至 (1)；

(3) 对该轮蚂蚁寻得的路径做单点交叉和变异算子操作，然后重新确定该轮循环中的最优路径；

(4) 按式(3.9)更新全局信息素；

(5) 如果达到结束条件，算法结束，否则跳至 (1) 继续执行。

流程图如下。

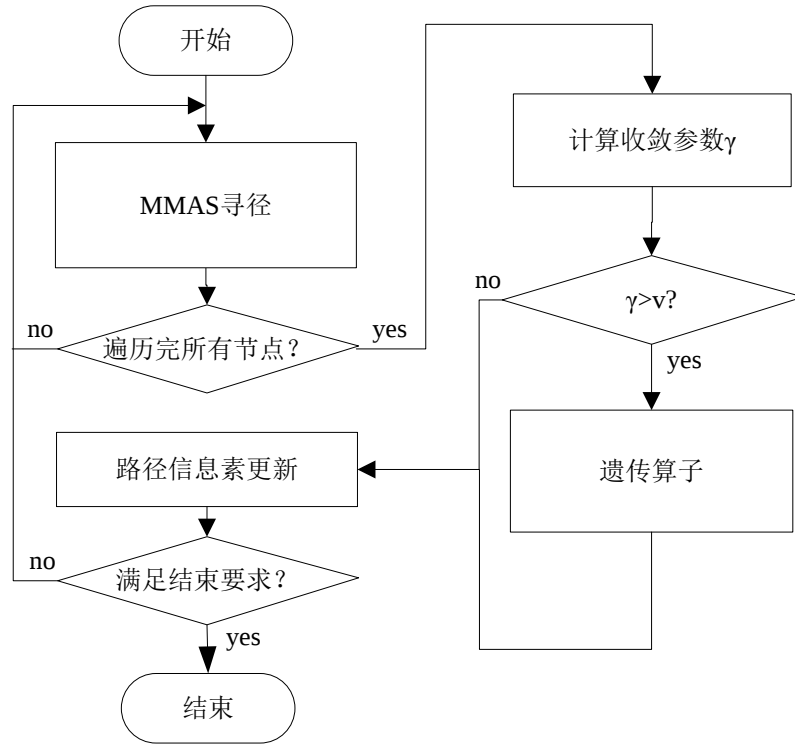


图 3.10 融合模块流程图

3.5 算法总体流程及框架

上面对本文提出的基于遗传和蚁群的自适应路径规划算法各个阶段的设计思想和方法进行了详细阐述。本节将对本文提出的算法的整体流程做一个整体的阐述。算法的详细流程如下：

- (1) 设置初始种群大小，遗传算子操作概率；
- (2) 初始化第一代路径种群；
- (3) 按式(3.3)对种群中的个体进行适应值计算；
- (4) 根据适应值对种群做选择、交叉、变异操作。产生下一代种群；
- (5) 对新老种群需采取精英选择策略；
- (6) 判断 GA 是否符合终止条件，如果是则跳至 (7)。否则跳到 (3) 继续执行；
- (7) 利用 GA 生成种群按式(3.7)设置蚁群算法初始引导信息素分布；
- (8) 按式(3.7)设置初始信息素分布，融合次数 $n = 0$ ，蚂蚁个数为 m ，初始化蚂蚁移动禁忌表；

华中科技大学硕士学位论文

(9) 将 m 个蚂蚁置于起点;

(10) 依次移动 m 个蚂蚁, 让它们按概率公式选择下一次的行动方向, 并更新蚂蚁的移动禁忌表。直到每个蚂蚁都到达终点, 同时记录每只蚂蚁的路径, 并确定当前的最优路径;

(11) 按式(3.11)计算该轮蚁群算法收敛速度参数 γ , 如果 $\gamma > v$ 跳至 (12), 否则调至 (13);

(12) 对该轮蚂蚁寻得的路径做单点交叉和变异算子操作, 然后重新确定该轮循环中的最优路径;

(13) 按式(3.9)更新全局信息素;

(14) 如果达到结束条件, 算法结束, 输出路径集, 否则跳至 (9) 继续执行。

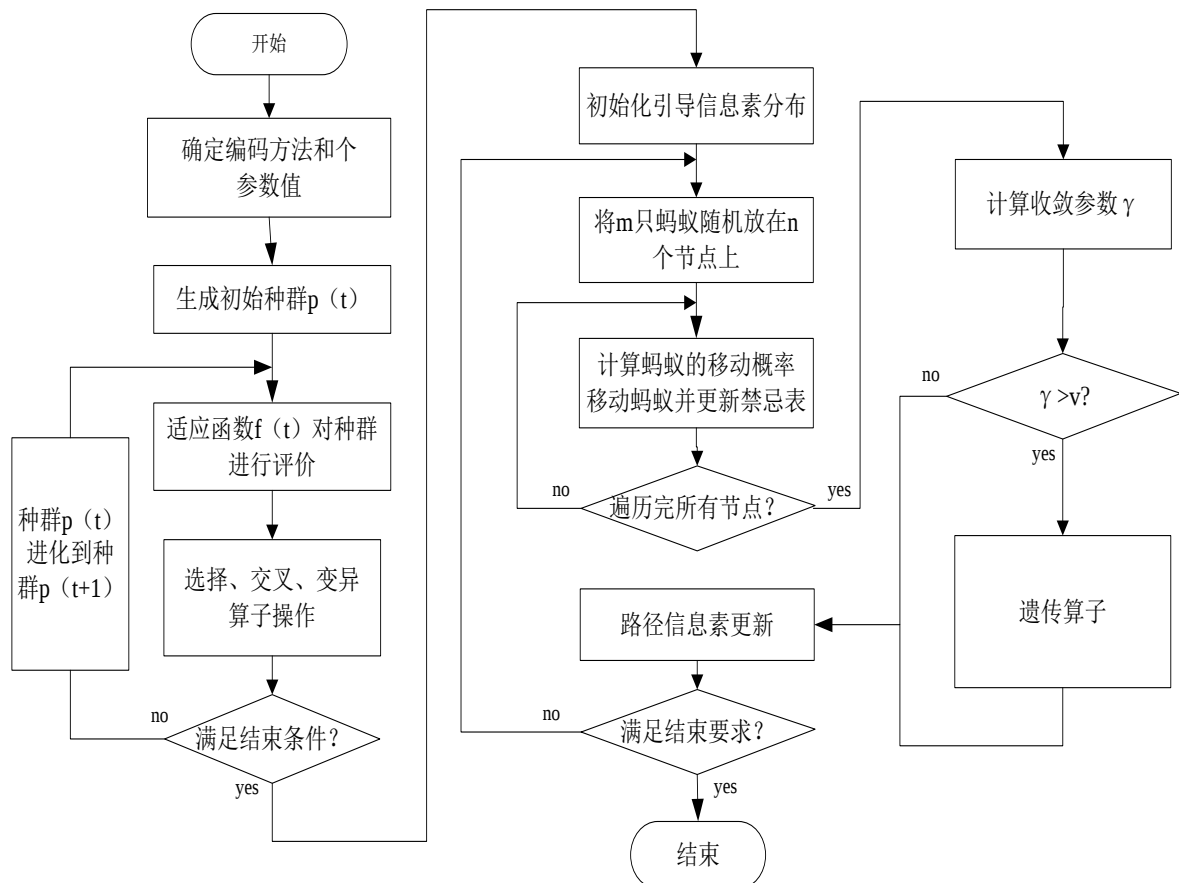


图 3.11 AGAAA 流程图

华中科技大学硕士学位论文

结合 AGAAA 的流程，本文算法的设计及实现总体框架如图 3.12 所示，在预处理阶段利用栅格法对寻径环境进行建模，然后利用改进的 GA 求解出一组解集作为 MMAS 算法的初始信息素分布，接着蚁群算法在此基础上进行路径规划，同时自适应融合遗传算子，以提高算法解的全局最优性，最后算法满足结束条件退出，输出最优路径。

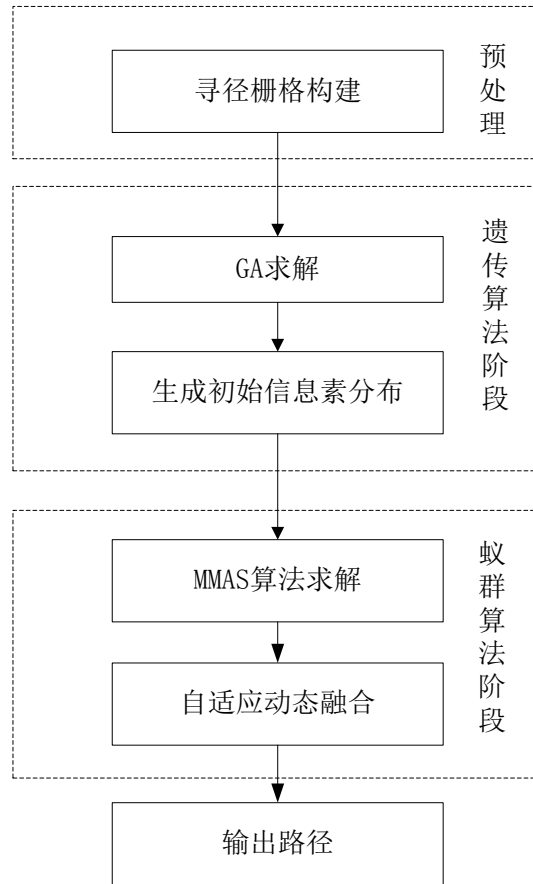


图 3.12 算法设计与实现总流程

3.6 本章小结

本章主要对基于 GA 和 MMAS 的自适应融合路径规划算法介绍。首先，介绍了前期 GA 针对路径规划的设计，然后提出了后期引入收敛速度参数的 MMAS 算法，它可以自适应的动态融合遗传算子，利用遗传算子的全局搜索性，在保证时间效率的基础上大大降低了后期蚁群算法由于收敛速度过快陷入局部最优的概率，使算法收敛与全局最优解。

华中科技大学硕士学位论文

4 算法实现及性能测试

基于遗传算法和最大最小蚂蚁算法的智能路径规划算法必须通过仿真实验来证明其可行性和有效性，以及确定各个参数的最佳组合方案。上一章对算法的设计思想进行了详细的介绍，在此基础上，本章给出了算法具体的仿真实现，概述整个仿真实验的环境和各个功能模块。并通过在不同寻路环境下的大量的实验数据来得到算法各个模块的参数组合最优方案，最后在相同参数的情况下通过和基本融合算法以及 A* 算法的仿真实验比较，证明了本文提出算法的性能优越性。

4.1 算法实验平台

为了验证本文提出算法的可行性、有效性以及性能上的优势。必须对其进行仿真实现。本节简单阐述了仿真系统的结构和内容以及算法的开发工具和运行环境。

4.1.1 仿真系统描述

在对算法进行的仿真实验中，在后台自动生成的寻径环境为长宽 20×20 的网状栅格，并设置寻径对象的起点为左上角栅格，终点为右下角栅格。前台利用 OpenGL 对寻路环境和最后寻得的路径进行绘制，环境中黑色栅格代表障碍物，白色栅格代表可行路径。为了对算法的可行性和有效性进行全方面的验证，仿真的智能路径规划实验在多种障碍物环境下共同进行实验论证。其中一些寻路环境中设有深度陷阱障碍，以确保本文设计的算法可以应对任何情况。

4.1.2 开发和运行环境

为了满足本文自适应动态融合算法对原有遗传算法和最大最小蚂蚁算法针对路径规划问题的设计和改进需求，我们采用 Dev-C++ 进行开发，同时也保证了算法运行的效率。

(1) 开发环境

Dev-C++ 4.9.9.2;

(2) 运行环境

操作系统: Microsoft Windows XP;

华中科技大学硕士学位论文

CPU: Intel(R) Core (TM)2 Duo 系列双核处理器, 主频 2.0GHz;
内存: 2G。

4.2 算法参数组合测试

众所周知, 遗传算法和蚁群算法是两种经典的人工智能方法。根据他们各自不同的优点, 这两种算法被应用于各个领域之中。由于算法中参数众多, 如何设置它们成为一个公开性的问题, 因为直接关系到算法的性能和运行的结果。但是由于两种算法变形应用于众多领域, 实际上目前为止尚没有一个公认的最佳参数标准。根据本文提出的路径规划仿真环境, 本节将在这一问题上通过大量实验数据来确定它们的最优参数组合。本节中的每条数据均为 20 次实验结果的平均值。

4.2.1 遗传算法路径规划参数测试

GA 中影响算法运行的参数众多, 其中最为关键的为种群大小、变异概率、交叉概率, 符号分别表示为 N_{pop} , P_{cross} , $P_{mutation}$ 。此外还将测试本文提出的适应度函数式(3.3)中的两种权重因子 α , β , 共 5 中参数对遗传算法结果和性能的影响。规定遗传算法结束条件中最小迭代次数 $G_{Min} = 20$, 最大迭代次数 $G_{max} = 100$, 最大停滞代数 $G_{Stop} = 5$, 最小进化率 $G_{ratio} = 1\%$ 。

(1) N_{pop} : 该参数过大会造成算法计算量急剧增大, 且后期过多冗余计算, 过小不利于种群的多样性, 导致算法收敛于局部最优。对于 N_{pop} 组中, 规定 $P_{cross} = 0.8$, $P_{mutation} = 0.05$, $\alpha = 0.5$, $\beta = 0.5$ 。

表 4.1 种群数量组运行结果

种群数量	平均适应度	最佳适应度	运行时间 (ms)
20	1.88	2.24	33.00
40	2.08	2.48	79.95
60	2.21	2.71	154.25
80	2.22	2.72	227.65
100	2.25	2.78	344.80
120	2.31	2.79	460.30

由表 4.1 可知随着种群中个体数量的提升, 种群的平均适应度和最佳适应度都有

华中科技大学硕士学位论文

所提升。同时，算法的运行时间也在增加。观察可得，种群数量到达 60 以后，算法的时间消耗大于路径适应度的提升值，所以我们选取 $N_{Pop} = 60$ 。

(2) P_{cross} ：该参数过大造成优秀个体染色体结构被破坏，过小则不利于种群收敛于全局最优解。对于 P_{cross} 组中，规定 $N_{Pop} = 60$ ， $P_{mutation} = 0.05$ ， $\alpha = 0.5$ ， $\beta = 0.5$ 。

表 4.2 交叉概率组运行结果

交叉概率	平均适应度	最佳适应度	运行时间 (ms)
0.4	2.12	2.46	165.15
0.5	2.07	2.49	156.40
0.6	2.14	2.56	146.35
0.7	1.98	2.36	142.10
0.8	2.24	2.76	140.60
0.9	2.15	2.62	156.40

由表 4.2 可得在前几组数据中，种群的最佳适应度和平均适应度随着交叉概率的增加而增加。当交叉概率增大到 0.8 以后，此两项指标则和交叉概率成反比关系，呈下降趋势，而在运行时间该项指标上没有明显的差别。所以我们选取 $P_{cross} = 0.8$ 。

(3) $P_{mutation}$ ：该参数过大使寻优过程趋于随机搜索，过小不能获得某段的局部最优从何更好的配合交叉算子完成全局搜索。对于 $P_{mutation}$ 组中，规定 $N_{Pop} = 60$ ， $P_{cross} = 0.6$ ， $\alpha = 0.5$ ， $\beta = 0.5$ 。

表 4.3 变异概率组运行结果

变异概率	平均适应度	最佳适应度	运行时间 (ms)
0.01	2.06	2.12	92.25
0.03	2.22	2.38	113.10
0.05	2.00	2.38	138.25
0.07	2.18	2.68	157.75
0.09	2.10	2.64	187.55
0.11	2.01	2.56	203.50

由表 4.3 可知 0.07 之前平均适应度随着变异概率的增大而增大，而之后则呈下降趋势。而算法的运行时间和变异概率一直呈正比关系。这里在保证最好的寻径准确性的前提下选取 $P_{mutation} = 0.07$ 。

华中科技大学硕士学位论文

(4) α, β : 两种参数的组合决定了算法对于个体解是否连通的重视程度, 由于本文提出的 GA 是作为整个融合算法的前期过渡部分, 所以对于路径解较优秀但是其中有个别障碍物的个体具有倾向性。适应度函数中 $\alpha + \beta = 1$, 其中 β 越算法越倾向于连通路程, 反之障碍物对个体适应值的影响越小。总而言之, 在适度的情况下, 本文提出的适应度函数的目的是筛选出优秀的可以指导蚁群算法搜索的个体, 而并不要求路径一定要连通。以下是在不同的参数搭配情况下的解路径。

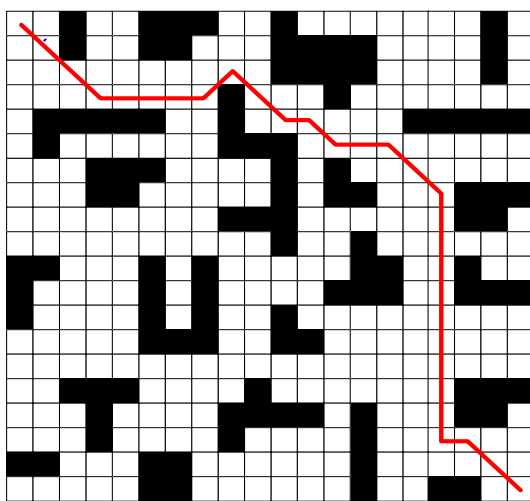


图 4.1 $\alpha = 0.1, \beta = 0.9$ 路径

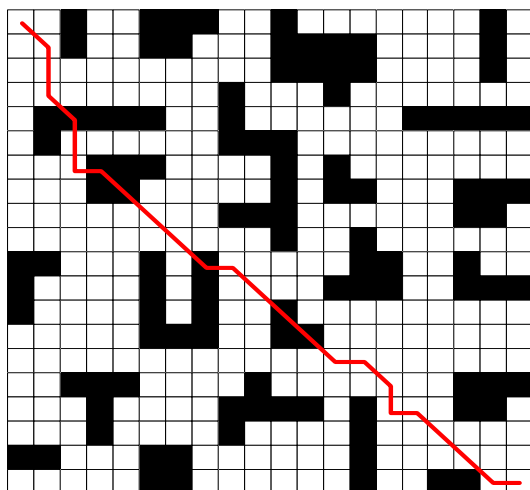


图 4.2 $\alpha = 0.9, \beta = 0.1$ 路径

华中科技大学硕士学位论文

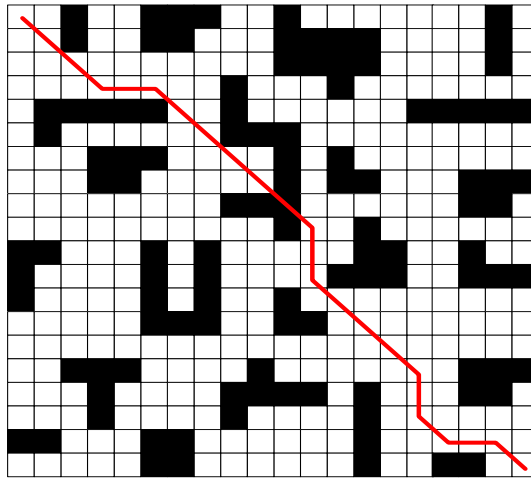


图 4.3 $\alpha = 0.5, \beta = 0.5$ 路径

由图 4.1 到图 4.3 可得，图 4.1 为了保证连通性而舍近求远。图 4.2 对障碍物的考虑几乎忽略，穿越的障碍物过多。图 4.3 在保证路径优秀的前提下，只有穿越一个障碍物，对蚂蚁算法的指导作用的影响几乎可以忽略，所以本文选 $\alpha = 0.5, \beta = 0.5$ 。

4.2.2 蚁群算法路径规划参数测试

MMAS 算法模型中最为主要的参数为：蚂蚁数量、信息素启发式因子、期望启发因子、信息素挥发因子、信息素浓度常数，直接影响算法的性能，分别用符号 m ， α ， β ， ρ ， Q 表示。算法最大迭代次数为 80 次，信息素区间设置为当 $[10, 100]$ ，最优蚂蚁的路径进化率连续 5 代小于 1% 时算法自动退出。

(1) m ：表示参与寻路的蚂蚁数量，该值过大会造成寻径环境信息素分布过于平均，正反馈作用不能得到充分体现。反之，该值过小可能导致大量路径没有机会被搜索，算法会陷入局部最优解。对于 m 组，规定 $\alpha = 5$ ， $\beta = 5$ ， $\rho = 0.7$ ， $Q = 50$ 。

表 4.4 蚂蚁数量组运行结果

蚂蚁数量	平均路径长度	最佳路径长度	运行时间 (ms)
5	86.00	58.16	699.15
10	36.82	31.71	203.20
30	36.77	31.67	422.85
50	36.41	31.62	520.45
70	36.30	31.43	715.65
90	36.50	31.67	811.90
110	37.27	31.99	1031.35

华中科技大学硕士学位论文

由表 4.4 可得在前几组数据中, 蚂蚁最佳路径长度和平均路径长度随着蚂蚁数量的增加而增加。当蚂蚁数量增大到 70 以后, 此两项指标则和蚂蚁数量成反比关系、所以我们选取 $m = 70$ 。

(2) ρ : 信息素会发因子, 表示每轮信息素的损耗。该值过大算法过早停滞收敛, 过小正反馈机制不明显。对于 ρ 组, 规定 $\alpha = 5$, $\beta = 5$, $m = 50$, $Q = 50$ 。

表 4.5 挥发因子组运行结果

信息素会发因子	平均路径长度	最佳路径长度	运行时间 (ms)
0.4	36.51	32.05	441.60
0.5	36.78	31.88	492.35
0.6	37.63	31.69	560.35
0.7	36.30	31.44	574.55
0.8	35.86	31.18	579.80
0.9	37.71	32.64	963.60

由表 4.5 可得在前几组数据中, 蚂蚁最佳路径长度和平均路径长度随着挥发因子的增加而基本呈增加趋势。当挥发因子增大到 0.8 以后, 此两项指标则和挥发因子呈反比关系、所以我们选取 $\rho = 0.8$ 。

(3) Q : 信息素浓度常数, 表示每轮信息素加强的信息素强度。该参数越大收敛速度越快, 反之越慢。对于 Q 组, 规定 $\alpha = 5$, $\beta = 5$, $m = 50$, $\rho = 0.7$ 。

表 4.6 信息素浓度常数组运行结果

信息素浓度常数	平均路径长度	最佳路径长度	运行时间 (ms)
10	36.51	31.40	995.55
30	36.47	31.69	694.85
50	37.50	31.82	578.35
70	36.13	31.80	558.15
90	37.50	31.82	488.60
110	35.64	31.01	528.30
130	36.56	31.48	556.30

由表 4.6 可知, 除了 110 组, 其它各组数据没有特别明显的变化。110 组表现为最优的指标, 有可能因为它是接近 MMAS 系统中最大信息素参数的一个数值, 所

华中科技大学硕士学位论文

以我们选取 $Q=110$ 。

(4) α, β : α 为信息素启发因子, 反映了在选择移动方向时信息素对选择概率的影响, 该参数越大, 正反馈效应越强。 β 为期望启发因子, 反应了选择路径长度对选择概率的影响, 该参数过大选择就会倾向于当前最短路径的贪心算法。对于 α, β 组, 规定 $Q=50$, $m=50$, $\rho=0.7$ 。

表 4.7 两种启发因子组运行结果

信息素启发因子	期望启发因子	平均路径长度	最佳路径长度	运行时间 (ms)
1	5	57.17	30.94	999.35
2	5	34.50	29.81	1047.70
3	5	35.49	31.20	912.55
4	5	35.57	31.03	743.70
5	1	40.26	32.72	588.10
5	2	40.25	32.77	989.80
5	3	37.91	31.97	984.55
5	4	38.35	31.73	757.30
5	5	36.88	31.90	750.00

由表 4.7 的组合因子数据可得, 第二组无论是在平均路径长度还是最佳路径长度上均表现最优, 通过分析各组的参数组合和结果数据可得, 且信息素启发因子大于期望启发因子时, 算法表现最佳。所以这里我们选取 $\alpha=2, \beta=5$ 。

4.2.3 融合模块参数测试

本节将对基于自适应融合的遗传和蚁群路径规划算法的两个融合部分的参数设定进行研究, 第一个部分为遗传算法结束后过渡到蚁群算法的过渡参数, 第二部分为蚁群算法内部自适应动态融合遗传算子的融合参数。

遗传算法过渡到蚁群算法的实质就是将 GA 的最终种群作为蚁群算法最初进行信息素进行强化的那一部分路径, 这样在一定信息素的指导下蚁群算法就能跳过初期信息素匮乏的阶段, 大大加快了收敛速度。所以在该部分中与 MMAS 算法的区别在于 MMAS 算法将所有路径点初始化信息素设为最大值, 而本文中的算法中如 3.3.1

节中的公式 $\tau = \tau_M + \tau_G$, 我们这里规定 $\tau_M = \tau_{\min} + \frac{1}{2}(\tau_{\max} - \tau_{\min})$, 也就是 $\tau_M = 55$ 。

华中科技大学硕士学位论文

表 4.8 加强信息素组运行结果

加强信息素	平均路径长度	最佳路径长度
5	38.27	38.55
10	38.32	32.85
15	37.88	32.76
20	37.24	32.32
25	35.27	30.64
30	37.56	31.84

由表 4.8 可得，当遗传算法加强的信息素为 25 时，算法的综合表现最佳。分析其原因可得，加强信息素量过小不能突出 GA 求解的指导作用，往往一条优秀路径在下一轮挥发过后信息素量的优势就不复存在，而过大就导致 MMAS 算法收敛过快，大大增加陷于局部最优的机率。所以这里我们选取 $\tau_G = 25$ 。

在蚁群算法自适应动态融合阶段如 3.4 节所描述，需要关注的参数有：遗传算子阈值，平均收敛率权重因子，最佳收敛率权重因子。分别用符号表示为 v 、 α 、 β 。

(1) v ：遗传算子阈值，当 MMAS 算法每轮迭代的收敛速度参数大于该阈值时，则进行一次路径交叉操作，该值过小会造成遗传算子操作急剧增加，不仅使算法运行效率下降而且会破坏优秀蚂蚁路径结构，过大则可能造成算法陷入局部最优的概率变大。我们规定另外两个参数 $\alpha = 0.5$ ， $\beta = 0.5$ 。

表 4.9 遗传阈值组运行结果

遗传算子阈值	平均路径长度	最佳路径长度	遗传算子操作次数
0.2	39.08	29.92	0.05
0.1	38.68	29.95	1.60
0.08	38.38	30.05	3.15
0.06	38.16	30.31	5.50
0.04	39.57	30.10	8.15
0.02	39.05	29.61	11.85
0.01	39.77	29.85	13.00

由表 4.9 可得随着遗传算子阈值降低，MMAS 算法中的遗传算子操作次数逐渐增高，最佳路径长度基本呈现变小的趋势，并在 0.02 达到峰值 29.61，几乎接近最优解。此后由于遗传算子操作过多，平均路径长度和最佳路径长度两项指标都有增大

华中科技大学硕士学位论文

的趋势。所以我们选择 $v = 2\%$ 。

(2) α, β : 两种参数的组合决定了算法对于整体或是个体进化的重视程度, α 较大则算法倾向于获得最优的蚂蚁, 相反, 算法则倾向于得到整体平均水平较优的解集, 这里我们规定 $v = 2\%$ 。

表 4.10 收敛速度权重参数组运行结果

α	β	平均路径长度	最佳路径长度
0.1	0.9	39.36	29.78
0.3	0.7	38.80	30.03
0.5	0.5	39.38	30.15
0.7	0.3	38.16	29.86
0.9	0.1	38.05	30.29

由表 4.10 可知各组数据之间波动不明显, 说明最优蚂蚁收敛速度和蚂蚁平均收敛速度内在有着很大的联系。但仔细观察可得, 当 α 最大是保证了平均路径长度为最优, β 最大保证了最佳路径长度为最优。综合考虑我们这里选择 $\alpha = 0.7, \beta = 0.3$ 。

4.3 算法性能测试

经过前面的工作, 已经得到算法每个阶段关键参数的最佳范围, 本节将在所有参数相同的情况下, 对基本融合算法、本文提出的自适应动态融合算法以及 A*算法在寻径的准确性和算法的性能上做一个比较。

4.3.1 测试案例

本文的测试环境为静态路径。众所周知, A*算法为当前静态环境下计算效率最高的算法, 被广泛运用于计算机游戏领域中, 所以本文选取该算法进行比较。根据上述说明, 本节将在多种障碍物环境下对以上 5 种算法在平均路径、最佳路径和运行时间三项指标上的表现进行比较分析得出最后的结论。与上一节相同, 本节中的每条数据均为 20 次实验结果的平均值。

4.3.2 结果分析

具体测试的实验结果分析如表 4.11 和表 4.12 所示, GAAA 表示基于 GA 和 MMAS 的基本融合算法, AGAAA 代表本文提出的自适应动态融合算法, A*表示 A 星算法。

华中科技大学硕士学位论文

由上一节的大量实验论证，这里我们规定 GA 中： $N_{pop} = 60$ 、 $P_{cross} = 0.8$ 、 $P_{mutation} = 0.07$ 、 $\alpha = 0.5$ 、 $\beta = 0.5$ ，MMAS 中： $Q = 110$ 、 $m = 70$ 、 $\rho = 0.8$ 、 $\alpha = 2$ 、 $\beta = 5$ ，自适应融合模块中： $\tau_G = 25$ 、 $v = 2\%$ 、 $\alpha = 0.7$ 、 $\beta = 0.3$ 。A*算法的启发函数取点到终点的欧式距离。

表 4.11 环境 1 试验数据测试结果

算法名称	平均路径长度	最佳路径长度	运行时间 (ms)
AGAAA	48.67	31.69	1894.60
GAAA	51.45	32.12	1821.52
A*	37.26	37.26	1012.82

表 4.12 环境 2 试验数据测试结果

算法名称	平均路径长度	最佳路径长度	运行时间 (ms)
AGAAA	32.98	29.21	1687.20
GAAA	39.32	33.26	1629.30
A*	39.12	39.12	1032.12

数据方面如表 4.11 和表 4.12 所示，在牺牲少许运行时间的条件下，AGAAA 寻径算法在平均路径和最优路径两项指标上的表现都明显优于 GAAA，而且在复杂环境下这一点被表现的尤为突出。另外，AGAAA 的平均路径和最优路径之间的差值明显小于 GAAA，可以得到在算法稳定性方面本文提出的算法也更胜一筹。可见，AGAAA 无论是在寻径的准确性和稳定性上都明显优于 GAAA 且更适用于复杂环境下的路径规划问题。

A*算法只适用于简单环境，其搜索范围是选择性的所以算法的时间效率很高，但是对于较为复杂的环境其寻路的准确性就会急剧下降。A*算法每次寻得的路径都是一样的固定路径，而且启发性函数的选择具有限制性，它只是理论上可以得到全局最优解的一种方法，在复杂环境下在寻路的准确性方面远远不如 AGAAA。

由图 4.4 和图 4.9 为 20×20 环境下三种算法寻得路径的对比。由图可知，不论是简单的寻径环境 1 还是复杂的寻径环境 2，AGAAA 都具有有效性和可行性。图 4.4 为简单环境下 AGAAA 最优路径，长度为 30.0416，图 4.5 为复杂环境下 AGAAA 最优路径，长度为 28.6274，两者都逼近于全局最优解。而另外两外两种算法在简单环境下表现尚可，距离最优解相差不远，但是在复杂环境下寻径准确性则有较为明显

的下降趋势。

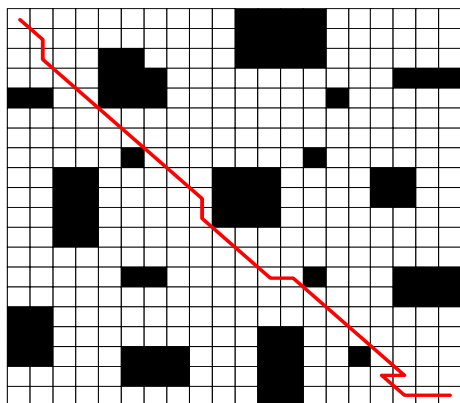


图 4.4 环境 1 下 AGAAA 路径

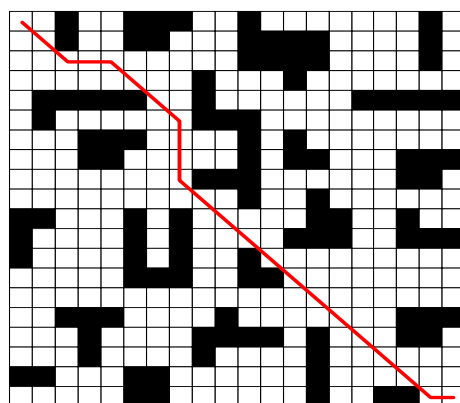


图 4.5 环境 2 下 AGAAA 路径

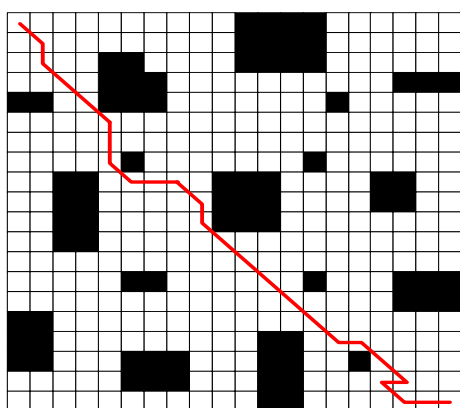


图 4.6 环境 1 下 GAAA 路径

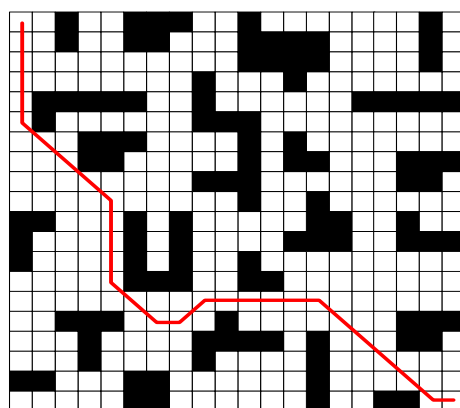


图 4.7 环境 2 下 GAAA 路径

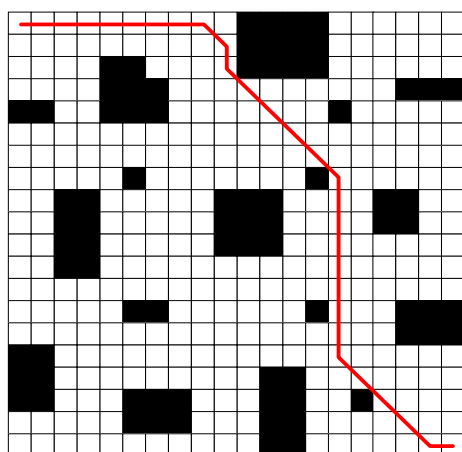


图 4.8 环境 1 下 A*算法路径

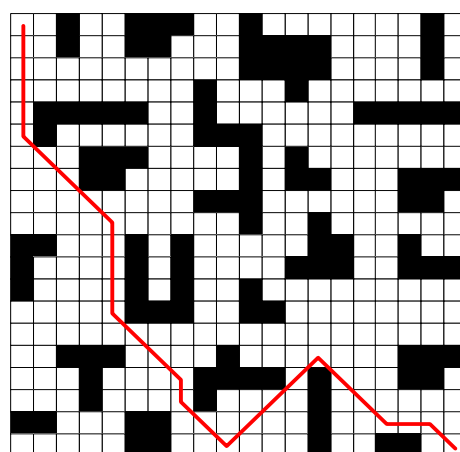


图 4.9 环境 2 下 A*算法路径

为了进一步证明算法在较大场景下的实用性,本文分别在 10×10 、 15×15 、 20×20 、 25×25 、 30×30 、 35×35 、 40×40 、 45×45 、 50×50 的复杂场景下做了对比试验,不同规模的场景复杂度与图 4.5 类似。场景规模和寻径准确性的关系如图 4.10 所示。

华中科技大学硕士学位论文

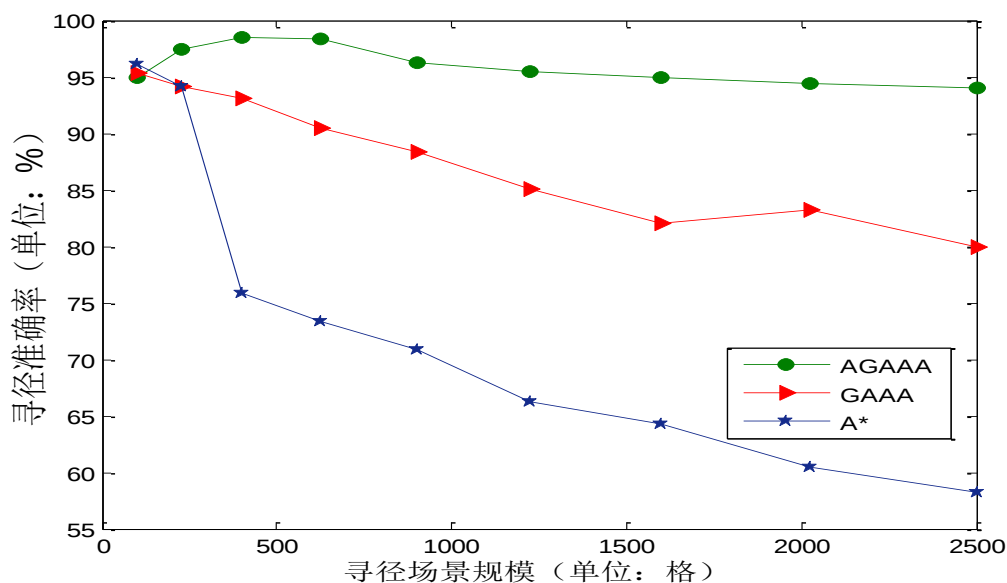


图 4.10 不同规模环境下算法寻径准确性

图 4.10 纵坐标中的寻径准确性为场景下最优路径和该算法求得路径长度的比值。由图可以看出随着场景的规模不断变大, AGAAA 基于趋于全局最优解, 在 25×25 的场景下达到峰值, 随后有稍许下降。GAAA 和 A* 算法的解都在不断偏离最优解, 其中 A* 算法随着场景规模的变大其寻路的准确性下滑幅度较大。

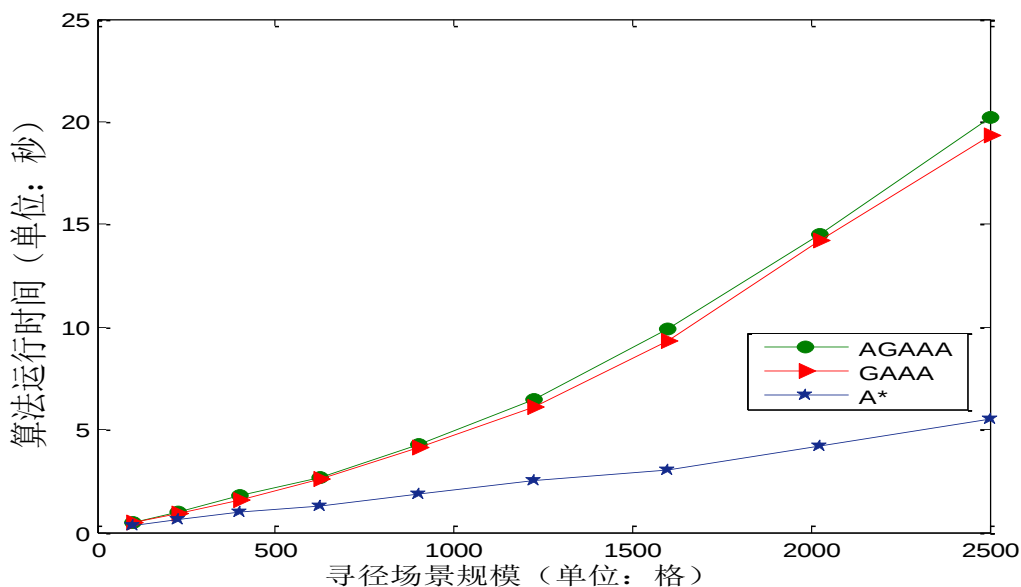


图 4.11 不同规模环境下算法运行时间

与图 4.10 中同样规模下三种算法的运行时间的比较如图 4.11 所示。我们可以得

华中科技大学硕士学位论文

到,在场景规模逐渐变大的情况下,A*算法由于它的选择性搜索,所以运行时间增长的不明显。而GAAA和AGAAA的运行时间明显高于A*算法,其中AGAAA由于后期融入了遗传算子操作时间略高于GAAA。

通过上述对比试验我们可以看出,与GAAA相比,在牺牲略微时间复杂度的前提下,AGAAA大大提高了算法的寻径准确性,使解可以收敛于全局最优,有效的降低了算法陷入局部最优的概率。且通过在不同规模场景下的仿真实验可以证明AGAAA应对大规模的场景时同样具有可行性,其寻路准确性基本保持在一个较高水平上,而A*算法在大而复杂的场景下表现则不尽如人意。但是由于A*算法是一种启发式搜索算法,它采用选择性搜索,所以时间效率较高,而AGAAA在时间效率上则随着场景的变大下降幅度较大。最后我们可以得出,本文提出的算法在寻径准确性方面表现突出,可以适应大而复杂的场景,但是计算时间与A*算法相比较慢。适合应用于一些实时性不高的应用,如智能机器人、道路导航等领域。而A*算法计算时间较快,但是面对各种复杂环境则显得力不从心,适用于一些人工设置好的虚拟简单场景,如计算机游戏领域。

为了进一步验证该算法的可用行,本文又另外在其他几种不同情形的障碍物环境下进行了可行性验证,最后得到本算法可适应大部分静态环境,且可在较短时间内计算出路径,寻径场景及路线如图4.12到图4.15所示。另外值得一提的是,在如图4.14的横向障碍物环境下,计算速度相比纵向环境较慢,这和遗传算法生成初始种群的方法有关,后续要对该问题进行研究,提出更为有效的方法。

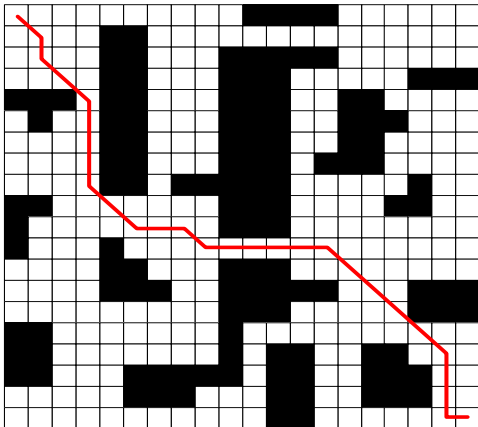


图 4.12 环境 3 下 AGAAA 路径

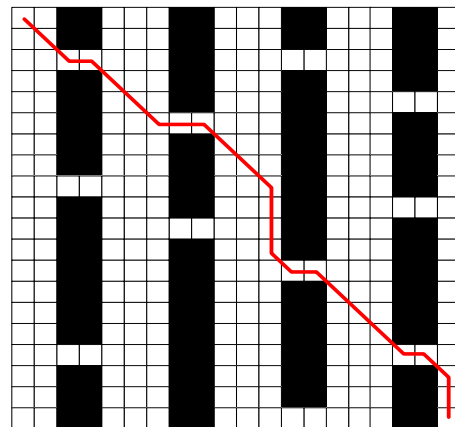


图 4.13 环境 4 下 AGAAA 路径

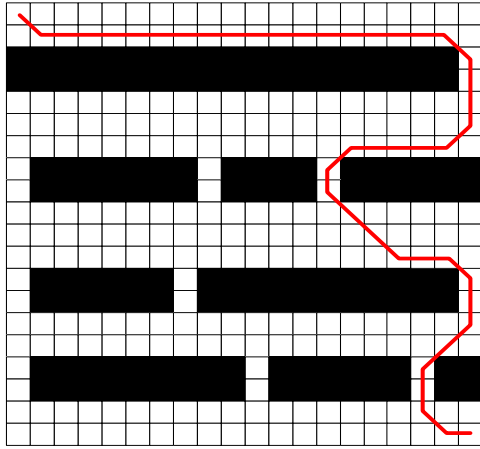


图 4.14 环境 5 下 AGAAA 路径

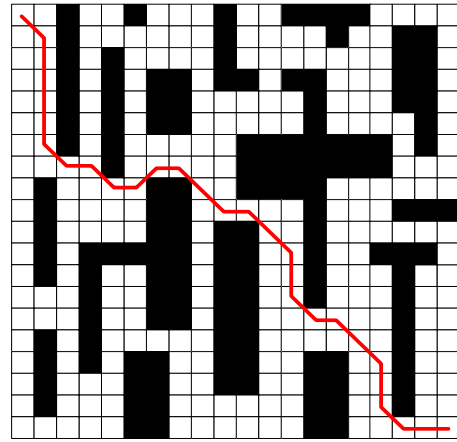


图 4.15 环境 6 下 AGAAA 路径

4.4 本章小结

本章利用 Dev c++作为基于遗传和蚁群的自适应动态融合路径规划算法的开发和测试平台，对算法进行实现和测试。并通过多种寻路环境下的大量的实验数据来确定算法各个阶段的参数组合最优方案，最后在相同参数的情况下。通过和基本融合算法以及 A*算法的仿真实验比较，证明了本文提出算法的性能优越性。

华中科技大学硕士学位论文

5 总结与展望

5.1 本文主要工作

本文致力于研究智能路径规划算法，总结了国内外学者关于路径规划算法的研究工作，提出了一种基于遗传算法和蚁群算法的自适应动态融合的方法，不仅提高了解的收敛速度，而且改善了后期蚁群算法收敛过快导致陷入局部最优的情况，增强了整个算法的全局搜索性。本文主要的研究内容如下：

(1) 对智能路径规划方法进行深入探讨，主要介绍了遗传算法和蚁群算法的发展，分析了它们的基本原理和算法步骤。提出了一种基于遗传和蚁群的自适应动态融合的算法。该算法在汲取遗传算法和蚁群算法不同阶段收敛速度优势的基础上加入了收敛速度参数，使得整个算法后期的蚁群算法可以自适应的融合遗传算子。在提高速度的基础上大大降低了算法陷入局部最优的概率。

(2) 在栅格法建模的基础上，设计并实现了本文提出的路径规划算法。并分别在多种障碍物环境两种情况下对该算法的可行性和有效性进行了验证。同时研究了本人设计的算法各个模块的最佳范围以及动态融合模块各个参数大小和协同关系的对算法的影响。

(3) 在相同参数的基础上同简单融合算法以及 A*算法在寻径的准确性和算法运算时间上做了比较。通过不同规模环境下的对比试验得出：与 GAAA 相比，在牺牲略微时间复杂度的前提下，AGAAA 大大提高了算法的寻径准确性，使解可以收敛于全局最优，有效的降低了算法陷入局部最优的概率。且通过在不同规模场景下的仿真实验可以证明 AGAAA 在应对大规模的场景时同样具有可行性，其寻路准确性基本保持在一个较高水平上，而 A*算法在大而复杂的场景下表现则不尽如人意。但是由于 A*算法是一种启发式搜索算法，采用选择性搜索，所以时间效率较高，而 AGAAA 在时间效率上则随着场景的变大下降幅度较大。最后我们可以得出，AGAAA 在寻径准确性方面表现突出，可以适应大而复杂的场景。但是计算时间与 A*算法相比较慢。适合应用于一些实时性不高的应用，如智能机器人、道路导航等领域。而 A*算法计算时间较快，但是面对各种复杂环境则显得力不从心，适用于一些人工设置好的虚拟简单场景，如计算机游戏领域。

华中科技大学硕士学位论文

5.2 进一步研究方向

路径规划是机器人、虚拟现实、计算机游戏领域研究重点之一。多年以来，国内外学者对于路径规划提出了许多方案。本文只是就智能路径规划算法给出了自己的一点看法，在原有算法的基础上进行了改进并予以实现。由于条件、时间的限制，还有很多后续工作需要进一步进行：

（1）本文提出的生成初始种群方法在应对横向的障碍物时运算速度较慢，研究一种更为有效的生成初始种群方法；

（2）利用遗传和蚁群两种算法的内在隐含并行性，设计并行算法。充分并行计算，对此类计算量较大的智能算法实行分而治之的计算，更充分的利用并行资源；

（3）遗传算法和融合模块中各个参数的内隐。

华中科技大学硕士学位论文

致 谢

时光匆匆，两年多的硕士生涯就要结束了。在攻读硕士学位期间，我得到了许多来自老师、同学和朋友的帮助，使我能顺利地完成学业，借此一隅，我要表达对他们的敬意和感谢。

首先，在两年半的研究生学业结束之际，我要感谢我的导师胡迎松老师。胡老师细心的工作态度、认真的工作作风和严谨的做人做事态度使我受益匪浅。并且在日常的工作和学习中，他都给予我无限的关怀和悉心地指导。不仅帮助我能够顺利地完成研究生的学业，而且使我学会了如何更好的做事做人。本文是在胡老师的严格要求和认真指导下完成的，从论文的选题、研究工作的开展到论文的撰写以及审阅与批改，胡老师投入了大量的心血。

感谢课题组的高鹏毅和李丹两位老师对我的指导和帮助，他们在我的待人接物以及工作态度和方法上都提出了很有价值的建议，在此表示深深的谢意。

在实验室学习期间，我得到了实验室同学莫大的帮助。感谢胡祥潇师兄给予的细心指导，他深厚的理论基础、丰富的工程经验和极强的动手能力给我树立了学习的楷模；感谢何青师姐在我加入实验室伊始，认真细致地教给我理论知识和科研方法，帮我尽快地融入到研究生学习和生活中；感谢蒋晶晶师兄在我遇到困难时给予的无私帮助，耐心细致地帮我解决了很多难题；感谢罗萧师兄，与他的交流让我对更广更深的科学领域有了初步了解，并萌发兴趣。

尤其感谢研究生期间的同窗好友李俊、肖军、董浩、赖明辉、王亮、孙文杰在学习和生活中给我很多帮助，增添了学习的动力，分享了生活的乐趣，一起经历了难忘的研究生生涯的点点滴滴，我在研究生期间取得的些许成果是与他们的通力协作分不开的。最后，深深感谢我的父母在我的成长过程中倾注了大量心血，我远离家乡在外求学，他们一直是我的坚实后盾，他们对我毫无保留的支持和关爱，使我有信心和勇气去面对未来的困难和挑战，给了我最大的动力和勇气。

最后，谨向百忙之中抽出宝贵时间评审本论文的专家、老师致以最诚挚的谢意。

华中科技大学硕士学位论文

参考文献

- [1] Naimi H M, Taherinejad N. New robust and efficient ant colony algorithms using new interpretation of local updating process. Expert systems with applications, 2009, 39(1): 481 ~ 448.
- [2] 任世军, 洪炳熔等. 一种基于栅格扩展的机器人路径规划方法. 哈尔滨工业大学学报, 2001, 33(1): 68 ~ 72.
- [3] B. Paden, A. Mees and M. Fisher. Path Planning Using a Jacobian-based Freespace Generation Algorithm. In Proceedings of IEEE International Conference on Robotics Automation, Scottsdale, USA, IEEE Press, 1989: 1732 ~ 1737.
- [4] A. Zelinsky. A Mobile Robot Exploration Algorithm. In Proceedings of IEEE International Conference on Robotics and Automation, Nice, France, IEEE Press, 1992: 707 ~ 717.
- [5] 慕延华等. 一种室内环境下移动机器人的路径规划新算法. 计算机工程与应用, 2006, 20(3): 58 ~ 60.
- [6] Ram A, Arkin R, Boone G. Using genetic algorithms to learn reactive control parameters for autonomous robotic navigation. Adaptive Behavior, 1994, 2(3): 277 ~ 304.
- [7] Brooks R. Solving the Find-path problem by good representation of free space. In Proceedings of IEEE International Conference on Sys Man and Cybern, New York, USA, 1983: 190 ~ 197.
- [8] Khatib. Real-time obstacle for manipulators and mobile robot. The international Journal of Robotic Research, 1986, 1(3): 90 ~ 98.
- [9] Ismail Chabini, Shan Lan. Adaptations of the A* Algorithm for the Computation of Fastest Paths in Deterministic Discrete-TimeDynamic Networks. In Proceedings of IEEE International Conference on Intelligent Transportation Systems, 2002, 3(1): 60 ~ 74.
- [10] Richard E. Korf, Weixiong Zhang, Ignacio Thayer et al. Frontier Search. Journal of the ACM, 2005, 52(5): 715 ~ 748.

华中科技大学硕士学位论文

- [11] Jianming Guo, Liang Liu, Qing Liu et al. An Improvement of D* Algorithm for Mobile Robot Path Planning in Partial Unknown Enviroment. In Proceedings of the 2009 Second International Conference on Intelligent Computation Technology and Automation, Washington, USA, IEEE Press, 2009: 94 ~ 397.
- [12] 姜桂艳, 郑祖舵, 白竹. 基于记忆机制的动态交通路径优化算法. 吉林大学学报, 2007, 37(5): 1043 ~ 1048.
- [13] J.J.Grefenstette, R.Gopal et al. Genetic algorithms for the traveling salesman problem. In Proceedings of the First International Conference on Genetic Algorithm, Pittsburgh, USA, 1985: 160 ~ 168.
- [14] Ibrahim Kushchu. Genetic Programming and Evolutionary Generalization. In Proceedings of IEEE International Conference on Evolutionary Computation, 2002, 6(5): 431 ~ 442.
- [15] Chang Wook Ahm, R.S.Ramakrishna. A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Population. In Proceedings of IEEE International Conference on Evolutionary Computation, 2002, 6(5): 566 ~ 579.
- [16] 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. 模式识别与人工智能, 2002, 15(2): 161 ~ 166.
- [17] 肖晓明, 陈志兴, 高平安. 动态确定基因数的遗传算法路径规划. 计算机应用研究, 2009, 26(7): 2469 ~ 2470, 2486.
- [18] Liu C Q, Jiang M H. Virtual obstacle concept for local mini-mum recovery in potential based navigation. In Proceedings of IEEE International Conference on Robotic & Automation, 2000, 2: 984 ~ 986.
- [19] Ashraf A. Kassim, B.V.K. Vijaya Kumar. A Neural Network Architecture For Path Planning. In Proceedings of International Joint Conference on Neural Networks, 1992, 2(2): 787 ~ 792.
- [20] Dmitry V. Lebedev, Jochen J. Steil, Helge Ritter. Real-time path planning in dynamic environments: a comparison of three neural network models. In Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2003, 4(4): 3408 ~ 3413.
- [21] Dorigo M, Maniezzo V, Colorni A. The ant system:Optimization by a colony of

华中科技大学硕士学位论文

- cooperating agent. In Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 1996, 26(1): 29 ~ 41.
- [22] Thomas Stutzle, Holger H Hoos et al. Max-Min ant system. Future Generation Computer System, 2000, 16(8): 889 ~ 914.
- [23] 吴斌, 史忠植. 一种基于蚁群算法的 TSP 问题分段求解算法. 计算机学报, 2001, 24(12): 2 ~ 6.
- [24] 陈峻, 沈洁, 秦玲. 基于分布均匀度的自适应蚁群算法. 软件学报, 2003, 14(8): 1379 ~ 1387.
- [25] Doitsidis L, Valavanis K P, Tsourveloudis N C. Fuzzy logic based autonomous skid steering vehicle navigation. In Proceedings of the IEEE International Conference on Robotics and Automation, 2002, 2: 2171 ~ 2177.
- [26] 张汝波, 郭必祥, 熊江. 基于遗传蚁群算法的机器人全局路径规划研究. 哈尔滨工程大学学报, 2004, 25(6): 724 ~ 727.
- [27] 刘玲, 王耀南, 况菲. 基于神经网络和遗传算法的移动机器人路径规划. 计算机应用研究, 2007, 2: 264 ~ 268.
- [28] Ceyda Oguz, M. Fikret Ercan. A Genetic Algorithm for Hybrid Flow-shop Scheduling with Mutiprocessor Tasks. Journal of Scheduling, 2005, 8(4): 323 ~ 351.
- [29] 陈华华, 郭晔, 杜歆. 基于改进型遗传算法的动态障碍路径规划方法. 传感技术学报, 2006, 19(2): 520 ~ 524.
- [30] 何大阔, 王福利, 张春梅. 基于均匀设计的遗传算法参数设定. 东北大学学报, 2003, 24(5): 309 ~ 411.
- [31] 朱庆保. 动态复杂环境下的机器人路径规划蚂蚁预测算法. 计算机学报, 2005, 28(11): 1898 ~ 1909.
- [32] 陈峻, 沈洁, 秦玲. 蚁群算法求解连续空间优化问题的一种方法. 软件学报, 2002, 13(12): 2317 ~ 2323.
- [33] 唐良, 方廷健. 基于改进蚁群算法的路径规划方法. 中国科学技术大学学报, 2009, 39(9): 980 ~ 983.
- [34] Fan X Y, Luo X, Yi S et al. Optimal path planning for mobile robots based on intensified ant colony optimization algorithm. In Proceedings of IEEE International

华中科技大学硕士学位论文

- Conference on Robotics, Intelligent Systems and Signal Proceeding, 2003, 1: 131 ~ 136.
- [35] Ding Y Y, He Y, Jiang J P. Muti-robot cooperation method based on the ant algorithm. In Proceedings of IEEE International Conference on Swarm Intelligence Symposium, Hangzhou, China, IEEE Press, 2003: 14 ~ 18.
- [36] 朱庆保. 复杂环境下的机器人路径规划蚂蚁算法. 自动化学报. 2006, 32(4): 586 ~ 593.
- [37] 黄翰,郝志峰,吴春国. 蚁群算法的收敛速度分析.计算机学报. 2007, 30(8): 1344-1353.
- [38] Zhengda Meng, Qinqi Chen. Hybrid genetic-ant colony algorithm based job scheduling method research of arc welding robot. In Proceedings of IEEE International Conference on Information and Automation, Nanjing, China, IEEE Press , 2010: 718 ~ 722.
- [39] 熊志辉, 李思昆, 陈吉华. 遗传算法与蚂蚁算法动态融合的软硬件划分. 软件学报, 2005, 16(4): 503 ~ 512.
- [40] 丁建立, 陈增强, 袁著祉. 遗传算法与蚂蚁算法融合的马尔可夫收敛性分析. 自动化学报, 2004, 30(4): 629 ~ 634.
- [41] 张汝波, 郭必祥, 熊江. 基于遗传蚁群算法的机器人全局路径规划研究. 哈尔滨工程大学学报, 2004, 25(6): 724 ~ 727.
- [42] 丁建立, 陈增强, 袁著祉. 遗传算法与蚂蚁算法的融合. 计算机研究与发展, 2003, 40(9): 1351 ~ 1356.
- [43] Shyi-Ming Chen, Chih-Yao Chien. Parallelized genetic ant colony systems for solving the traveling salesman problem. Expert Systems with Applications, 2011, 38(4): 3873 ~ 3883.
- [44] Zhang Qiang, Lei Humin, Wang Fei et al. Research of genetic-ant colony fusion algorithm for multi-missile cooperative combat target assignment problem. In Proceedings of the 21st annual international conference Chinese control and decision conference, Guilin, China, IEEE Press, 2009: 5940 ~ 5942.

华中科技大学硕士学位论文

附录 攻读学位期间发表论文目录

- [1] Yingsong Hu, **Yin Ding**, Dan Li, A Self-adaptive Path Planning Algorithm Based on Genetic and Ant Colony Algorithms, In the Proceedings of 4th International Conference on Information Technology and Services, Yantai, 2012(已投).