

G · 格子达论文检测报告【全文标注】

报告编号: DFF412DDB9EB44D0B8EB382C7DE24CA3

作者: 徐鸿飞

送检时间: 2021-06-07 01:06:34

送检单位: 中国地质大学(武汉)

送检文档: 机器人全局路径规划方法研究

对比索引检测检测范围:

1989-01-01至2021-06-07

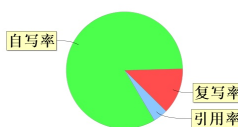
学术期刊库 近百亿+	学位论文库 13亿+	会议论文库 7000万+	报纸资源库 900万+	互联网资源库 数千万+
共享资源库 数百亿+	本科论文共享库 近亿+	大学生同届库 实时更新	专利库 近亿+	格子达论文库 30亿+

检测结果:

总相似比: 16.9%

论文总字数: 29546个

检测指标:



自写率: 83.1%

复写率: 12.81%

引用率: 4.09%

其他类型检测结果: 去除引用后总相似比: 12.81% 同校同届总相似比: 0.59%

其他指标:

图件	表格数	引用句子数	相似片段数
22	25	9	66

复写率来源:	学术期刊: 0.31%	学位论文: 0.36%	会议论文: 0.29%
	报纸资源: 0.42%	互联网: 1.61%	大学生本科论文: 9.82%
	共享资源库: 0.0%	专利库: 0%	

复写相似文献列表:

序号	相似文献	相似片段数	相似字数	相似比
1	来源: 大学生本科毕业论文联合共享库 出处: 2018	31个	1273	6.63%
2	来源: 大学生本科毕业论文联合共享库 出处: 2020	2个	87	0.45%
3	篇名: AR导航能成为“路痴”福音吗 来源: 报纸资源库 作者: 代小佩 出处: 科技日报 2018	1个	81	0.42%
4	来源: 大学生本科毕业论文联合共享库 出处: 2020	1个	69	0.36%

序号	相似文献	相似片段数	相似字数	相似比
5	来源:大学生本科毕业论文联合共享库 出处: 2020	1个	65	0.34%
6	来源:大学生本科毕业论文联合共享库 出处: 2018	1个	65	0.34%
7	来源:大学生本科毕业论文联合共享库 出处: 2020	1个	57	0.3%
8	篇名:ROS笔记(22) Gmapping_氢键H-H-CSDN博客_gmapping算法原理 来源:互联网文库 出处:http://www.baidu.com/link?url=QSpbMNU7r10B8HoP7...0	1个	56	0.29%
9	来源:大学生本科毕业论文联合共享库 出处: 2020	1个	54	0.28%
10	来源:大学生本科毕业论文联合共享库 出处: 2020	1个	51	0.27%
11	篇名:信息熵原理在电力系统最优潮流中的应用研究 来源:学位论文库 作者:李洪勤 出处:硕博学位论文 2005	1个	51	0.27%
12	篇名:基因突变_百度百科 来源:互联网文库 出处:http://www.baidu.com/link?url=nIkDGJqjJ4zBBpC8y...	1个	47	0.24%
13	来源:大学生本科毕业论文联合共享库 出处: 2019	1个	47	0.24%
14	来源:大学生本科毕业论文联合共享库 出处: 2019	1个	43	0.22%
15	篇名:...多目标和单配送中心的冷链物流配送路径优化研究——以... 来源:互联网文库 出处:http://www.baidu.com/link?url=vo9SzVl48K6SLGAGu...0	1个	41	0.21%
16	来源:大学生本科毕业论文联合共享库 出处: 2018	1个	39	0.2%
17	篇名:基于遗传算法的公路纵断面优化研究 - docin.com豆丁网 来源:互联网文库 出处:http://www.baidu.com/link?url=j7sMGJqjJ4zBBpC8y...	1个	36	0.19%
18	篇名:基于多项式与遗传算法的智能车辆换道轨迹规划研究 来源:会议论文库 作者:郑艳 商雄 许龙 出处:第29届中国控制与决策会议 2017	1个	37	0.19%
19	篇名:电路划分问题的遗传算法求解 来源:学术期刊库 作者:南国芳 李敏强 寇纪淞 出处:南国芳 李敏强 寇纪淞 2004	1个	37	0.19%
20	篇名:v智能控制第8章 遗传算法_百度文库 来源:互联网文库 出处:http://www.baidu.com/link?url=83W6cyd69yfHHQRU1...	1个	35	0.18%
21	篇名:...oftheapplicabilityofgeneticalgorithmstospace... 来源:互联网文库 出处:http://www.baidu.com/link?url=Kj-jd-k_B0zsgHiYF...0	1个	29	0.15%
22	篇名:7.1.1导航模块简介_赵虚左-CSDN博客 来源:互联网文库 出处:http://www.baidu.com/link?url=1Th4vNyIMoE4bxzfJ...0	1个	27	0.14%

序号	相似文献	相似片段数	相似字数	相似比
23	篇名:Neural network and genetic algorithm based global path planning in a static environment 来源:学术期刊库 作者:杜歆 陈华华 顾伟康 出处:杜歆 陈华华 顾伟康 2005	1个	23	0.12%
24	篇名:...robotbasedonK-meansalgorithm--《Transactionsof... 来源:互联网文库 出处:http://www.baidu.com/link?url=kx4cGoSM3u-xjKCvQ...0	1个	21	0.11%
25	来源:大学生本科毕业论文联合共享库 出处: 2020	1个	20	0.1%
26	篇名:南京地区一次大雾过程的多普勒天气雷达回波研究 来源:会议论文库 作者:徐姝 魏鸣 出处:2009第五届苏皖两省大气探测、环境遥感与电子技术学术研讨会 2009	1个	19	0.1%
27	篇名:信用评估与信用卡欺诈侦测的智能决策系统研究 来源:学位论文库 作者:李睿 出处:硕博学位论文 2011	1个	19	0.1%
28	来源:大学生本科毕业论文联合共享库 出处: 2020	1个	17	0.09%
29	篇名:基于改进遗传算法的移动机器人路径规划的研究-青海大学 来源:互联网文库 出处:http://www.baidu.com/link?url=Hfx-FhfdpjzGHgKNL...0	1个	17	0.09%

同属相似文献列表:

序号	学院	院系专业	相似文献	相似片段数	相似字数	相似比
1	中国地质大学(武汉)	材料与化学学院应用化学(卓越计划)	篇名:买尔外提-论文-液质定量分析沉积物中的药物残留	1个	22	0.11%
2	中国地质大学(武汉)	地理与信息工程学院地理空间信息工程	篇名:20171000730-房祖锐-变化环境下鄱阳湖流域径流变化模拟与归因研究	1个	21	0.11%
3	中国地质大学(武汉)	机械与电子信息学院电子信息工程	篇名:20171000292-万欣-基于遗传算法的极化码构造算法研究	1个	22	0.11%
4	中国地质大学(武汉)	地理与信息工程学院遥感科学与技术	篇名:基于卷积神经网络和弱监督学习的目标检测	1个	17	0.09%
5	中国地质大学(武汉)	自动化学院自动化	篇名:本科毕业论文-韩晨-2021.5.26	1个	17	0.09%
6	中国地质大学(武汉)	自动化学院测控技术与仪器	篇名:20171001751-陈远群-仓储物流移动机器人路径规划的应用研究	1个	15	0.08%

全文内容:

学士学位论文
机器人全局路径规划方法研究
二〇二一年六月

中国地质大学（武汉）学士学位论文原创性声明

本人郑重声明：本人所呈交的学士学位论文《机器人全局路径规划方法研究》，是本人在指导老师的指导下，在中国地质大学（武汉）攻读学士学位期间独立进行研究工作所取得的成果。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，对论文的完成提供过帮助的有关人员已在文中说明并致以谢意。

本人所呈交的学士学位论文没有违反学术道德和学术规范，没有侵权行为，并愿意承担由此而产生的法律责任和法律后果。

学位论文作者签名：

日 期： 年 月 日

摘要

自2016年工业和信息化部、国家发展改革委、财政部等三部委联合印发了《机器人产业发展规划（2016-2020年）》并指出为贯彻落实好《中国制造2025》将机器人作为重点发展领域的总体部署以来，我国机器人就已经进入了飞速发展时期，家用型机器人进入了千家万户。路径规划作为机器人技术中的一项重要研究内容被无数学者关注。

本课题以室内自主移动机器人和遗传算法路径规划为背景，首先阐述了遗传算法的基本原理和流程以及几种常用遗传算子与优化方法，并基于遗传算法提出一种新的全局路径规划流程，在已知的有障碍物的栅格地图中采取几何空间路径规划算法为机器人生成一条初始路径，并比较了几种不同的方法的生成速度与质量；再利用遗传算法对传统算法生成的初始路径进行优化，进一步提高路径在地图环境中的适应度值；并对折线路径进行曲线优化，使用两种不同的方法将折线转换为贝塞尔曲线，以获得运动效率最好的目标路径。最后为了验证上述算法的有效性，在ROS环境中使用Gazebo搭建仿真环境，将算法作为插件注册到ROS中并进行仿真实验。

根据模拟实验与Gazebo中的仿真结果，此方法用作路径规划速度快且用时较短，能够运用在机器人路径规划上且具有比较好的效果。

关键词：遗传算法；路径规划；机器人；贝塞尔曲线

Abstract

Since 2016, the Ministry of Industry and Information Technology, the National Development and Reform Commission, and the Ministry of Finance jointly issued the "Robot Industry Development Plan (2016-2020)" and pointed out that in order to implement the "Made in China 2025", robots will be the key development Since the overall deployment of the field, our country's robots have entered a period of rapid development, and domestic robots have entered thousands of households. As an important research content in robotics, path planning has attracted the attention of countless scholars.

Based on the background of indoor autonomous mobile robots and genetic algorithm path planning, this topic first explains the basic principles and processes of genetic algorithm and several commonly used genetic operators and optimization methods, and proposes a new global path planning process based on genetic algorithm. In the known grid map with obstacles, the geometric space path planning algorithm is used to generate an initial path for the robot, and the generation speed and quality of several different methods are compared; the genetic algorithm is used to generate the initial path generated by the traditional algorithm. Carry out optimization to further improve the fitness value of the path in the map environment; to optimize the curve of the polyline path, use two different methods to convert the polyline into a Bezier curve to obtain the target path with the best motion efficiency. Finally, in order to verify the effectiveness of the above algorithm, Gazebo was used to build a simulation environment in the ROS environment, the algorithm was

registered as a plug-in in ROS and the simulation experiment was performed.

According to the simulation experiment and the simulation results in Gazebo, this method is fast and short in path planning, and can be used in robot path planning with relatively good results.

Key Words: Genetic algorithm; path planning; robot; Bezier curve

图清单

- 图 2.1 机器人底盘结构图5
- 图 2.2 XBot-U机器人示意图6
- 图 3.1 遗传算法流程图10
- 图 3.2 轮盘赌示意图14
- 图 4.1 模拟程序所用地图17
- 图 4.2 简单路径示例图18
- 图 4.3 使用贪心+回溯的方法生成的初始路径图21
- 图 4.4 使用单次深度优先搜索生成的初始路径22
- 图 4.5 两种曲线化方法对比图26
- 图 4.6 迭代次数与适应度值图27
- 图 4.7 未曲线优化前的路径图28
- 图 4.8 曲线优化后的最终路径图28
- 图 5.1 软件技术博物馆模型图29
- 图 5.2 机器人XBot-U30
- 图 5.3 使用Gmapping SLAM构建地图33
- 图 5.4 仿真环境中使用Gmapping生成的地图33
- 图 5.5 在ROS中的节点通信图33
- 图 5.6 使用rviz设置机器人目标姿态33
- 图 5.7 在ROS中使用遗传算法进行全局路径规划图34
- 图 5.8 使用parrot_planner用作路径规划35
- 图 5.9 使用navfn用作路径规划35
- 图 5.10 使用本文的算法用作路径规划36

表清单

- 表 4.1 不同方法生成初始路径对比22
- 表 4.2 不同迭代次数情况27
- 表 5.1 环境参数表30
- 表 5.2 不同起始点路径规划结果表34
- 表 5.3 相同起始点不同路径规划算法效果对比35

绪论

章节标识符(隐藏)

选题背景和意义

随着科学技术日新月异的进步，我国已经步入经济转型的拐点区间，很多人力工作都实现了自动化，很多行业都已经开始了机器人替代人工、生产工艺升级换代的步伐，移动智能机器人也出现在了越来越多领域，家用机器人也变得越来越普遍。

自主移动机器人是一类能够通过携带的多个传感器感知环境状态和自身状态，实现在有障碍的环境中向着目标自主运动，从而完成指定功能的机器人系统[1]，具有很强的自规划、自组织、自适应能力。而全局路径规划算法就是自主移动机器人技术中一项重要研究内容，移动机器人的移动能力和效率是依赖于路径规划方法的，从起点到目标点，一个好的路径规划方法可以使得两点之间的路径长度变短、平滑且安全，路径短可以减少机器人的移动时间，路径平滑可以减少转弯增加移动效率，路径路径安全则保障了机器人在移动过程中不会发生事故，因此，研究全局路径规划算法对于移动机器人来说具有非常重要的意义。

传统路径规划方法主要是图搜索算法，通过某种方式来遍历搜索图，虽然可以找出路径最短的路径，但是不一定适用于机器人，如迪杰斯特拉算法[2] 则更适用于计算机网络中的最短路径搜寻，机器人的路径规划还需要考虑到更多的约束条件，如安全距离、运动效率等。

本课题提出了一种新的全局路径规划算法流程，在已知的有障碍物的栅格地图中采取几何空间路径规划算法为机器人生成一条初始路径；再利用遗传算法对传统算法生成的初始路径进行优化；并对折线路径进行曲线优化，以获得运动效率最优或次优的目标路径。

研究现状分析

移动机器人的全局路径规划是指如何在已知地图与定位的、有障碍物的作业场景中找到从指定起点到终点的最优或次优路径，使得机器人在运动过程中可以安全且无碰撞地绕过环境中的所有障碍物，其实质就是机器人在运动过程中的导航和避障。

其中机器人的地图与定位目前一般使用SLAM来完成，SLAM (Simultaneous localization and mapping) 的全名是同时定位与地图构建，是一种概念：机器人在未知的环境中，通过各种传感器获取到的数据来确定自身的位置，同时将各种数据综合增量式的构建所处场景的地图。而在运行时不需要构建地图而只需要获取定位时，一般则使用AMCL来获取定位。AMCL[3] 全称为Adaptive Monte Carlo Localization，蒙特卡洛自适应定位是用于2D移动的机器人的概率定位系统。它实现了自适应（或KLD采样）蒙特卡洛定位方法，是一种比较常用的定位算法，通过使用粒子过滤器比较已知地图和检测到的障碍物来进行定位。

目前主要的全局路径规划算法有：图搜索算法、人工势场法、模拟退火算法、神经网络算法、遗传算法等。

图搜索算法主要有DFS (Deep-First Search深度优先搜索)、BFS (Breadth-First Search广度优先搜索)、Dijkstra (迪杰斯特拉算法)、A* (A星算法) 等，核心思想是通过不同的搜索方法搜索地图来寻找最短路径；人工势场法[4] 通过在目标点周围构建引力场与在障碍物周围构建斥力场，再在这个叠加场中搜索出最优的路径，可以实现实时处理且生成的路径较为平滑，但是当障碍物变多时，容易造成混乱[5]；模拟退火[6] 算法通过模拟固体退火来实现在一个很大的搜寻空间内寻找一个最优解，此算法容易陷入局部最优解，但是它的微调能力很强；神经网络算法[7] 是一种模仿人脑神经网络的智能算法；遗传算法是一种全局优化的搜索算法，主要通过模拟达尔文进化论中的选择、杂交和变异的过程来不断的对生成的路径进行改进，这个算法也是本文主要研究的方法。

国内外研究现状

遗传算法最早由Holland教授及其学生提出[8]，模拟自然界中“物竞天择，适者生存”的法则由父代得到子代的迭代，最后从在最终种群中取得最优解，该算法拥有很好的自组织和自学习性，且可以很好的与其他的算法相结合使用，实现简单，但是同时也存在局部搜索能力差、容易陷入局部最优解、迭代次数多等问题。针对这些不足，使用遗传算法对移动机器人路径规划也受到了国内外学者的广泛研究。

在路径种群初始化方面，焦合军等直接使用A*算法来初始化路径[9]，胡章芳等则提出朝着目标点行走，遇到障碍物使用SPS算法产生点集，使用迪杰斯特拉算法测试产生的点集是否可行的方法[10]，M. Srinivas和L. M. Patnaik提出了一种概率自适应遗传算法[11]，使交叉和变异的概率会随着个体适应度值的变化而变化，减少了遗传算法迭代的次数从而更快的得到全局最优解。翁理国等在这个基础上提出了此方法等缺陷，遗传概率上限为事先给定的固定数值，不能根据种群的进化情况随时调整，限制了算法的收敛速度，然后引入了种群多样性的自适应调整[12]。王雷等提出一种改进的路径规划遗传算法[13]，首先是基于概率来生成初始路径，使用经营者策略来进行选择，同时也使用了自适应变异概率，将局部规划算法和全局规划相结合制定了碰撞策略，这种路径初始化方法速度快且具有较好的初始基因。杨嘉等在适应度函数中加入了路径平滑度[14]，并研究了路径长度和平滑度权重系数对算法的影响，优化之后可以有效的减少迭代次数，但是实验中使用的开始与终点必须在边缘有较大的局限性；Robotics提出了一种改进的基于差分进化混合多目标骨架粒子群算法的机器人路径规划方法[15]，在适应度上考虑路径长度、平滑度和路径安全度，通过改进的差分进化变异策略生成可行路径，同时提出一种新的帕累托控制方法，通过路径碰撞的可能性来确定路径点的最佳位置；Changfu Yang等针对遗传算法生成路径安全距离不足的问题，将人工势场法与遗传算法相结合[16]，提出了根据地图复杂度自适应添加或删除路径节点的自适应算子和监督算子；刘洋[17]和Jianwei Ma[18]等都提出了将贝塞尔曲线与遗传算法相结合来解决节点冗余和拐点角度大的问题，宋宇[19]、段立伟[20]和魏彤[21]等则是通过引入删除和插入算子来解决连贯性问题，而李国洪等则是通过B样条差值来实现[22]；Hao Kun[23]等和胡章芳等都提出了可以并行计算的小生

境法[10]，将一个大种群随机分成几个种群数目相同的小种群，有效的改进了遗传算法的性能。

可见很多学者在遗传算法路径规划的问题上，都是从初始化路径、增加特殊的算子、使用自适应变异和交叉概率与路径平滑等方面着手来进行优化的。

存在的问题

大部分学者使用遗传算法路径规划的研究都是使用 MATLAB这类软件进行模拟，且使用的是很小的（如10×10）的地图来进行实验，而在实际应用中则很少会在这种很小的地图上做全局路径规划，没有考虑到现实中地图大小带来的时间消耗，所以本文拟在一个和实际应用中大小相似的地图上研究基于遗传算法的路径规划，然后在一个实际的地图中进行仿真实验验证算法的可行性。

本文主要研究内容

本文主要研究机器人在基于SLAM与AMCL的技术下已知地图与自身位置的情景中研究如何利用遗传算法，图搜索算法等进行最优的路径规划，需要充分考虑障碍物和计算的时间消耗。本文结构安排如下：

第一章主要介绍本课题的选题背景以及研究意义，对国内外研究现状做了综述，并介绍了本文的各章节安排。

第二章主要介绍本文研究对象（机器人），以及在进行路径规划之前数据（地图）的获取，包括SLAM和AMCL技术，因为机器人要进行路径规划的前提就是要知道周围的环境（地图）与自己在哪儿（定位）。

第三章主要阐述遗传算法的机制，介绍了遗传算法的原理与流程，包括编码、初始化、选择、交叉、变异的一些做法。

第四章对遗传全局路径规划算法进行分析，并比较了几种不同的算子在路径规划中的使用情况，以及对优化后的折线路径再进行曲线优化。

第五章介绍了仿真实验平台，主要在ROS环境中使用Gazebo对算法进行仿真测试，验证遗传算法全局路径规划在实际应用中的可行性。

研究对象与数据

章节标识符(隐藏)

研究对象

本课题主要研究内容为机器人全局路径规划，所以主要研究对象为可以移动带有传感器的机器人，机器人的转向由差速驱动，可以实现原地转向，转向半径基本等于自身半径。

差速驱动是一种由两个轮子的移动系统，每个轮子都有独立的控制单元，主要由两个轮子的转速差来实现转向，当两个轮子向相反方向时即可原地转向。这种两轮机器人环境适应性强且运动灵活，适合任务环境复杂或变化较大的场合，其模型如图 2.1 机器人底盘结构图所示。

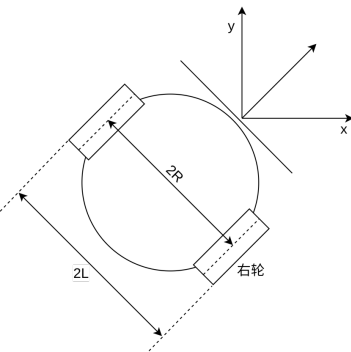


图 2.1 机器人底盘结构图

机器人底部盘为圆形，两侧分别有一个轮子，两个轮子共同控制机器人的运动，当两轮速度相同时，机器人则直线前进；当产生差速时，即为转弯，如向右转时为左轮速度大于右轮速度，原地转向则为两轮方向相反，速度相同。所以此机器人可以以任意角度转向，可以严格按照路径规划所得的路径进行行走，在通过路径规划算法获取到机器人路径后不断将达到下一要求位置的行进路径曲率半径及行进旋转角传给机器人，完成机器人的转向和前进。

本文实验中使用的二轮机器人模型为由重德智能与中国科学院软件研究所成立的智能机器人联合研究中心出

品的XBot-U教研科学机器人，它的结构如图 2.2 XBot-U机器人示意图所示。

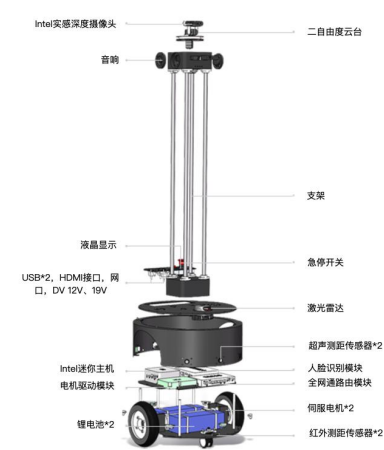


图 2.2 XBot-U机器人示意图

数据获取

本文的模拟实验主要在ROS系统中进行，ROS (Robot Operating System) 是一个应用于Linux系统中、开源的服务框架，所以特别适用于机器人这种多节点多任务的应用场景，广泛应用于机械臂、无人车等许多种类的机器人上。

本实验的平台是Ubuntu16.04，使用对应的ROS是Kinetic版本，ROS主要有以下特点：

分布式：ROS中每个功能对应一个节点，而不是所有功能放在一起，一个节点异常不会影响到其他节点的执行；

多种语言支持：ROS框架可以与多种流行编程语言结合，如常用的C++、python和Lisp；

插件式开发：ROS中指定了很多的接口，开发人员可以方便的集成这些接口生成插件注册到ROS当中；

开源社区：ROS拥有一个活跃的开源社区 (<https://wiki.ros.org/>)，使用人数多，开源的包也多，可以有效的问题。

本文使用的仿真工具为Gazebo，Gazebo是一个机器人3D动态模拟器，能够在准确的模拟出复杂的室内或室外场景以及高效的模拟机器人的运动。与游戏引擎类似，Gazebo提供了物理模拟引擎来模拟运动场景，同时也提供多种传感器以及直观的程序界面，对用户很友好。Gazebo与ROS一样也是开源的，同时在多个仿真工具中也是对ROS兼容最好的。

在ROS系统中，全局路径规划的数据主要有两个：一个提前准备的地图数据和一个运行时定位数据和给定的目标点数据。

机器人想要进行路径规划的前提就是需要知道地图与自身的位置，如上一章所介绍使用SLAM，Gmapping算法[24]是目前使用激光雷达和里程计以及基于粒子滤波方案里面相对成熟和可靠的一个SLAM算法，采用RBPF的方法效果稳定，许多使用ROS系统的机器人都使用了Gmapping作为SLAM算法，它被封装在了ros-perception组织中的slam_gmapping仓库中。本文主要就是使用Gmapping Slam算法来构建地图，最后构建的地图是一种pgm格式的栅格地图。本文中机器人运行的环境为中科院软件所提供的一个软件技术博物馆模型。

得到地图之后需要的就是运行时定位数据和给定的目标点数据了。目标点是人为指定的所以可以直接获取到（本文在ROS系统中使用Rviz设置机器人目标姿态程序中即可获取），本文也使用上一章节所介绍的AMCL算法来获取机器人在已知地图中的定位，它被封装在ROS中的Navigation Stack这个metapackage中。

所以本文会在路径规划前使用Gmapping Slam来构建一个完整的地图并在路径规划运行时使用AMCL算法来定位，这两个算法包都已经在ROS中进行了封装。

遗传算法原理与流程

章节标识符(隐藏)

遗传算法概述

根据达尔文进化论的“物竞天择，适者生存”[25]，自诞生以来，地球上的每个物种都经历了漫长的进化过程，在这个过程中，物种要生存下去就要去进行生存斗争，这包括与自然界的斗争与物种内部的斗争，经过不断的繁衍、斗争，具有更强的生存能力的个体就在自然选择中存活下来，并产生更多的后代，而相对的生存能力弱的就会被自然选择所淘汰，后代也会越来越少，直至完全消亡。

孟德尔和摩根认为[26]，产生这种现象的原因是因为个体的遗传物质不同，而遗传物质是作为一种指令编码以基因的形式存在与个体的DNA之中，每个基因都控制着个体所表现出来的在不同方面特性，不同的基因组合产生的个体所表现出来的特性就不一样，对自然环境的适应性也不一样，所以通过基因的杂交就有可能产生更强大的后代。

物种进化机制基本可以分为以下三种形式来运作：

自然选择：就是考验物种的生存能力，对自然环境适应性越强的具有更强的生存能力，越可能存活下来，进而产生后代，使得这种优良基因得以保留；

杂交：杂交是指父代两种不同的基因排列进行了交叉排列，使得物种产生结合两个个体优良基因的可能；

变异：变异指的是父代产生的子代染色体上某个基因进行了随机的改变，产生了完全不同于父代的个体，这使得物种具有了多样性，让进化不会有了终点。

很多学者对关于生物进化和遗传的机理进行了研究，并从中发展出适用于现实世界的复杂适应系统研究的计算技术：自然进化系统的计算模型和模拟进化算法，遗传算法就是由其中的Holland及其学生首先提出的一个重要的发展方向[27]。

遗传算法模拟了物种进化的过程，它通过将问题的解进行编码，并为每个解分配一个适应度值，然后重复的进行选择、交叉、变异，不断的迭代之后，在最后的群体中得到一个全局最优解或次优解。

遗传算法适用于全局最优化的搜索场景，是进化算法的一种，它与传统的爬山算法这种启发式优化算法相比，能够跳出局部最优而找到全局最优解，遗传算法只需定义好适应度函数，就可以在全部解空间内去进行搜索，拥有优秀的自组织性和自学习性[28]。

遗传算法流程

遗传算法主要流程如图所示，它模拟了生物种群进化的流程，假设有一个待解决的问题，一个典型的遗传算法的步骤（如图 3.1）为：

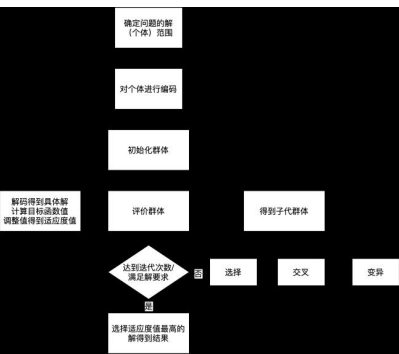


图 3.1 遗传算法流程图

首先是要定义问题解的范围，之后在初始化与进行迭代时产生的父代和子代都不能脱离解的范围，其次是确定遗传时的一些参数；

然后确定编码方式，对物种个体，也就是一个解进行编码，一般常用的是二进制编码，这样交叉与变异的时候更好操作；

初始化种群，在第一步定义的解的范围内随机生成指定种群数量的解并使用上一步确定的编码方式对解进行编码；

定义适应度函数，主要用于评价个体在这个问题中的好坏，对应物种个体在自然环境中的生存能力；

根据第一步确定的参数以及上一步的适应度函数计算得到的适应值，进行选择、交叉、变异的遗传操作生成子代，通常把这些遗传操作叫做遗传算子；

判断是否到达迭代次数或者解是否满足问题的要求，若满足则选择子代中适应度最高的解作为最终结果，若不满足则继续将子代作为下一次迭代的父代继续进行第五步的操作。

算法参数的确定

在遗传算法中，种群大小、交叉概率、变异概率等参数的取值是影响算法性能和效果的一个重要因素，因此这些参数的确定是遗传算法中是及其重要的，一般要首先确定这些参数的值，当然也有可能需要经过不断的实验测试才能确定效果更好的值。一般有如下几个参数：

位串长度，这个参数是可选的，因为有些问题使用的可能是不定长的位串，如本文研究的路径规划问题，不定长的位串的性能会更加好，但是一般这个参数是和问题的精度相关的，精度要求越高则位串越长；

种族个数，这个参数是必须的，它规定了初始群体以及之后每次迭代的群体的个数，这个值越大，种群的多样性越好，可以防止提前收敛，但是也会增加计算量，一般这个值的范围为[20, 200] [29] ；

交叉概率，控制交叉算子对任意两个父代进行交叉操作的次数，交叉概率越高，则生成的新的结构个体也越多，丢失存在的优良基因的概率也越大，但是也更有机会生成结合父代双方优良基因的子代，一般这个值的范围在[0. 6, 1] [30] ；

变异概率，控制变异算子变异对个体位串每一位进行变异操作的次数，这个参数值越大，则种群的多样性越高，且容易产生新的优良基因，但是同时也越容易使得优良基因被丢失。一般这个值的范围为[0. 005, 0. 01] [31] 。

这些参数都是与实际问题的密切相关的，在面对不同类型的问题时，都需要经过不断的测试才能确定对对应问题比较友好的参数值，特殊问题可能还有其他参数视问题而定。

遗传编码与种群初始化

遗传编码一般都是在解决问题时第一步需要确定的问题，它定义了遗传迭代时的个体与问题实际解的转换方式，把问题解转换为种群个体就是编码操作，反之则为解码操作。一般来说，遗传编码都会向着更方便的进行遗传操作这个方向靠拢，其中最常用的就是二进制编码，即将问题的解表示为由0, 1构成的位串，其他常用的编码方式还有：适用于TSP问题的序列编码、自适应编码、大字符集编码等。

初始化种群的个体一般是随机产生的，对于具有解空间范围约束的问题，有时也需要根据实际问题来具体实现，可能需要判定随机初始值所对应的解是否在解空间范围内，也可能是使用某一些方法来生成，以使得初始种群就具有较好的适应性，且同时保持种群的多样性，从而让遗传算法更快的收敛。

适应度函数

遗传编码将问题的解编码为种群个体的染色体位串空间，为了模拟自然界种群的“物竞天择、适者生存”的原则，需要为不同的染色体的适应度进行评价。所以，适应度函数就是在对应的生存环境中，根据个体的表现型计算出它的适应值，这个值就决定了个体在这个环境下的生存能力，一般适应值高的个体就具有更强的生存能力，也就是该个体对应的解更优。

由于适应度函数是个体在遗传操作过程中唯一的生存指标（不包括其他特殊算子（如删除算子）时），因此适应度函数的定义直接决定了种群的进化方向。根据实际问题的含义，适应值一般与路径长度、收入、时间联系在一起，且一般将适应值规定为非负，且越大则解越优。所以有时可以直接从目标函数得到，有时需要对目标函数进行一定的处理才能得到适应度函数。一般有如下几种处理方式，设目标函数为 $f(x)$ ，适应值函数为 $g(x)$ ：

适应度函数有结果为负的情况：

		(3. 1)
--	--	--------

其中 c 为一个指定的与实际问题的实数，可能经过测试才可以得出一个比较好的结果。

目标值最小化问题，一般有两种方法，第一种为取反：

		(3. 2)
--	--	--------

第二种为取倒数：

		(3. 3)
--	--	--------

通过这种转换之后，适应度函数就可以满足非负且值越大则解越优的条件了。

遗传算子

标准的遗传算法一般都包括选择、交叉、变异这三个基本的遗传算子，它们分别对应了自然界物种进化中的自然选择、杂交繁殖和基因突变的现象。遗传算法利用遗传算子来描述群体如何进化，所以遗传算子是遗传算法的重要组成部分，也是控制进化过程的基本工具。一般常用的有以下几种：

选择算子

选择算子，又叫做复制算子，即从父代种群中按照计算出的适应值来挑选哪些个体可以保留，而另一些被淘汰

汰的过程，即自然界中生物自然选择的过程。一般最常用也是最基本的选择方法为适应值比例选择，选择过程体现了达尔文进化论中“物竞天择、适者生存”的思想，保证了生存能力强的个体更大几率的生存下来并将优秀的基因遗传给下一代。对于规模为n的群体，假设适应值函数为g，即个体的适应值为，则其选择概率为：

$$P=a_1,a_2,a_3,\ldots,a_n$$

		(3. 4)
--	--	--------

得到每个个体的选择概率，即适应度值的和进行归一化之后，通常使用轮盘赌方法去选择，也称为比例选择或比例复制。通过对产生的随机数落在的区域所在的个体进行选择，类似赌场之中的轮盘（如图 3. 2）。

在实际代码中则是为每个个体计算一个区间：

		(3. 5)
--	--	--------



图 3. 2 轮盘赌示意图

然后随机生成一个0-1的随机数，判断这个数在哪个区间内则选择那个区间对应的个体，重复执行这个过程，直到选择的个体数量达到了要求为止，个体可以被重复选择，也有个体可能未被选择到，即被淘汰了。

交叉算子

交叉算子是模拟自然界中物种杂交基因重组过程的一种具有原始性的操作，它使得父代的优良基因得以结合传给下一代，以生成具有更优秀的基因，即适应度更高的新个体。交叉操作的步骤一般为：

从父代中随机选取两个准备进行交叉操作的个体；

假设位串的长度为L，对要交叉的个体随机选取[1， L-1]之间，即非两端的位置的一个或多个位置进行交叉操作，

随机[0, 1]生成随机数，小于交叉概率则在第二步生成的交叉点上进行交叉操作，两个个体交换各自的部分基因串，然后形成两个新的个体。

根据步骤2的生成交叉点的数量，交叉算子又可以分为单点交叉和多点交叉。一般单点交叉更加简单，操作的信息量更小，假设有两个待交叉的位串：

		(3. 6)
--	--	--------

随机生成的交叉点位置为，则进行交叉操作之后的新个体为：

$$k,1 \leq k \leq l$$

		(3. 7)
--	--	--------

但是在实际问题中，交叉可能受多方面因素的控制，随机交叉后的新个体可能不符合问题的解的要求，这时就需要使用与问题相对应的交叉方式，可能需要考虑编码的一些性质。如在TSP（旅行商人问题）问题中，路径点不能随意变化，它要求每个点只能经过一次，如果按照单点随机交叉就会出现重复经过某些点的问题，这时就可以选用选取基因点的方式来进行交叉操作，先从第一个位串中选取一些点，再从第二个位串中选择剩下的点进行填补，再把第一个位串中剩下的点填补到第二个位串中，这就避免了多次访问同一个点的问题，生成的新个体依然满足每个点只经过一次的要求。

变异算子

在遗传算法进化过程中，变异算子主要适用于增加群体的多样性，让群体不再受初始群体的基因的限制，它

给群体往更优的方向进化提供了无限的可能。它模拟了自然界中个体基因组DNA分子发生的突然的、可遗传的突变，从而染色体的结构也被改变的现象。在二进制编码遗传算法中，变异算子通过变异概率按位反转某为等为基因的值来实现。对于要变异的染色体：

$$P_m$$

		(3. 8)
--	--	--------

变异后的新个体为。在大部分遗传算法中，变异的概率都是比较小的，这也符合自然界中物种发生基因突变概率比较小的事实情况，因为变异概率较小，在变异过程中，很可能有些个体一个位变异都没有发生，造成了计算资源的浪费，所以在实际问题中，通常采用另一种和此方法效果相同的方法，先计算个体会发生变异的概率，当确定了会发生变异之后再对个体进行变异操作，步骤如下：

$$S = a_1, a_2, a_3, \dots, a_l$$

先计算个体会变异的概率，假设原始基因变异概率为，群体大小为n，则个体发生变异的概率为：

$$P_m$$

		(3. 9)
--	--	--------

则只有产生的随机数时，才会对进行变异操作。这时每个位变异的概率就不是了，需要使得总的变异的期望次数和原先的相等。原先方式的变异基因位数量为，设新的位变异概率为，则新方式的变异基因位数量为，需要让两者相等，则：

$$\text{rand}() < p_m(s_i)$$

		(3. 10)
--	--	---------

解得：

		(3. 11)
--	--	---------

然后将要变异的个体每位按照此概率进行位反转操作即可，降低了原先计算的复杂度。

在实际问题中，很多问题使用的可能并不是二进制编码，这时就需要根据实际问题来制定变异的方法，如上述的TSP问题的变异，他的编码就是旅行城市编号的顺序，随机变异的话又会导致重复经过同一座城市的问题，这时采用按照概率随机交换两个经过城市的序号的位置即可，既保证了经过城市的唯一性，也保证了遗传算法群体的多样性。

判断迭代何时终止

判断迭代何时终止需要看具体问题的要求，一般有如下几种方式：

最简单的也最常用的是采用规定最大迭代次数的方法，达到迭代次数之后即停止，但有时可能不够准确；

对于解有精度要求的问题，一般采用判断种群最优个体是否满足精度要求来决定是否终止迭代，此时的解已经满足了要求了；

对于对时间有限制的问题，就按照程序的运行时间是否到达限制来判断；

就是判断种群的收敛程度，如所有个体都一样了肯定要终止迭代了，这时只有变异算子在发挥作用了。

基于遗传算法的全局路径规划算法研究

章节标识符（隐藏）

根据第三章的遗传算法的流程，本章将把概算法应用到机器人全局路径规划这个实际问题。本章拟采用python构建一个简单500×500的地图环境，如图 4. 1，并在这个简单的地图上来测试遗传算法的表现，蓝色为起点（50, 100），绿色为终点（400, 400）。

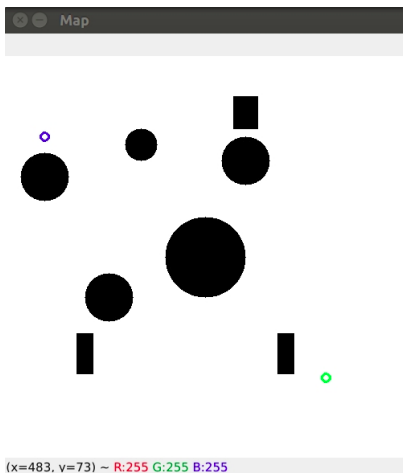


图 4.1 模拟程序所用地图

路径编码

路径编码可以将地图转换为一个一维数组，路径就是一系列的整数值，从而可以编码成二进制的串，但是这里并不采用这种方式，而是直接保留点的坐标，使用一系列的二维点作为路径，且使用的是不定长编码，因为不知道路径要经过多少个点，如果定死了经过点的个数，太少了可能找不到路径，太多了会增加问题的复杂度，如图 4.2。既是编码，也直接是问题的解，且利于后面进行交叉与变异，因为交叉与变异都不会使用二进制编码的方法。

`[(50,100), (130,180), (210,180), (290,180), (400,400)]`

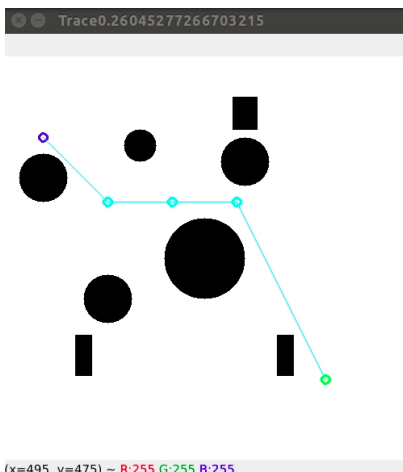


图 4.2 简单路径示例图

路径初始化

确定编码之后，首先要做的事情就是初始化群体，这也是整个流程中极其重要的一步，要求既要保证种群的多样性，同时也要具有优秀的基因以使得算法可以更好更快的收敛。对于路径规划这个问题来说，初始化路径一般有两种方向：

第一种是随机生成初始路径，路径可以经过障碍物，然后在定义适应度函数时给予经过障碍的路径以很大的惩罚；

另一种就是初始化的时候就直接生成不经过障碍物的初始路径，在选择、交叉、变异的过程中也不经过障碍物。

本文使用第二种方法，直接生成可用的初始路径，因为本文的方法是对一条初始路径进行优化，且机器人的路径规划一般都是即时响应的，这可以使得初始路径就可以携带更多的优秀基因，从而可以加快算法的收敛速度，减少迭代次数，获得更高的执行效率，对应的后续的遗传操作也无法使用随机性操作，而反观第一种方法，随机生成路径具有很大的不确定性，在后续进行的遗传算子操作中，也有可能生成不可用的路径，收敛速度缓慢，迭代次数大，执行效率低。

贪心+回溯生成初始路径

首先使用的方法是步长为1的贪心回溯，步骤如下：

初始化一个列表path，并将起点加入到path，

判断path[-1]是否为终点，若是将path拷贝一份作为一个初始个体存储，删除最后一个点后进入下一步，若不是，则直接进入步骤3，

寻找path[-1]周围的八个点中的可用点，即不是障碍物的点，将点按照距离终点的距离排序，按照距离终点更近的优先选择之前未选择过的点且path中不存在的加入path，若周围没有可用的点，则删除path中最后一个点重复步骤3；

一直执行这个回溯直至个体的数量达到要求停止。

但是经过测试之后，这个方法的效果并不好，原因是此操作是寻找邻居点，相当于每次搜索的步长为1，导致算法执行的速度很慢。

步长为1这种近似与全局搜索的方法复杂度很高，所以在选择邻居点时可以不选择临近的，而是直接选择指定距离（步长）的点作为邻居点，这时步长的确定成了一个问题，因为需要从起点到终点，起点到终点的x轴差与y轴差的绝对值需要是步长的倍数，所以选择x轴差与y轴差的绝对值的最大公约数作为步长，且在寻找可用邻居点时使用动态步长，即在距离为1时没有邻居点时，就寻找距离为l的邻居点。

$$\left\lfloor \frac{l}{2} \right\rfloor$$

这里引出一个新的问题，如何判断邻居点可达，之前距离为1的时候直接判断邻居点是否障碍物即可，但是这里邻居点与当前点之间是有距离的，因为是栅格地图，所以这里需要检测当前点与邻居点连线经过的所有栅格点是否有障碍物点，问题则转换为求两点连线之间经过的所有点，假设两个点坐标分别为，方法如下：

$$(x_1, y_1), (x_2, y_2)$$

首先判断与是否都为0，若是则两点重合，经过的点就是自身；

$$\Delta x = x_1 - x_2$$

判断与的大小，此处假设，则说明两点之间连线的点的所有x轴的值都不同；

$$\left\lfloor \Delta x \right\rfloor$$

直线上时对应的栅格点

（round为四舍五入取证函数，，此处不可能为0，因为若为0，则两点重合）

反之若，则将上一步的x，y互换即可，若，则都可行。

$$\left\lfloor \Delta x \right\rfloor < \left\lfloor \Delta y \right\rfloor$$

这样做虽然速度有所改进，但还是有可能很长时间只能生成固定数量的初始路径，这是因为它对每个点的邻居点都进行了搜索，虽然加入了优先级，但是当需要的初始路径数量多的时候，此方法还是近似于全局深度优先搜索，生成的路径也并没有携带很好的基因，路径的多样性不高。生成的路径虽然全都不一样，但是大部分初始部分都一样（如图 4.3），因为所有的路径都是根据最开始的选择得来的，它们可能都有一样的初始选择，只有在接近终点的地方才会有一些不一样，这样做的对后续的遗传算法效果是很差的。

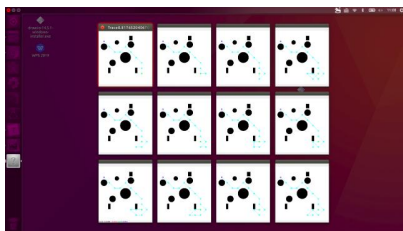


图 4.3 使用贪心+回溯的方法生成的初始路径图

基于轮盘赌的单次深度优先搜索

这里直接使用一个基于概率选择的单次深度搜索，模拟一个机器人在寻路，步骤如下：

初始化一个列表path，并将起点加入path；

获得path[-1]即当前点的可用且不在path中（避免重复）的邻居点并按照距离从小到达排序，假设这些点为，为每个点分配被选中的概率：

$$P_0, P_1, P_2, \dots, P_n$$

使用轮盘赌的方法对这些点进行选择，即生成随机数，若，则选择作为下一个点加入path；

$$p \in [0, 1]$$

重复2-3步骤直至无法获得有效邻居点退出或到达终点将path加入初始种群；

重复上述步骤直至初始种群数量达到目标值。

使用此方法生成的初始路径质量较高，且具有多样性。因为是根据概率来选择，保留了大部分局部最优的选择同时也不会陷入局部最优。

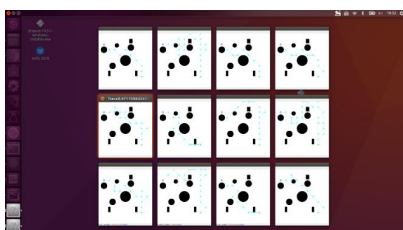


图 4.4 使用单次深度优先搜索生成的初始路径

如图 4.4，可见生成的路径多样性丰富，且都是可行路径。

使用上述的方法生成的初始路径质量已经很高了，但还是存在一个问题，若x轴差与y轴差的绝对值的最大公约数是一个很小的数，当最大公约数小至1后，则又会变成直接搜索临近点了，步长的改进就失去了效果。同时可以想到，当机器人所在的点与终点之间没有障碍物时，之后的搜索就没有意义了，这时可以选择直达终点而不是继续寻找邻居点，这样做之后步长就不需要使用x轴差与y轴差的绝对值的最大公约数了，因为寻找邻居点时是变长的，所以可以直接选择一个比较大的值，从而可以加快初始化路径的速度。

使用上述算法，以初始种群数量200测试如表 4.1 不同路径初始化方法生成初始路径对比所示。

表 4.1 不同路径初始化方法生成初始路径对比

方法	最短路径长度	平均路径长度	生成时间(s)
步长1回溯	-	-	>120
步长改进回溯	574.26	845.63	2.99
轮盘赌单次dfs	544.97	941.84	2.13
目标点直达优化	503.55	643.73	0.83

可以看见使用轮盘赌单次深度优先搜索加上目标点直达优化之后，种群初始化的效率获得了较大的提高，且生成的初始路径质量更好，具有了更好的初始基因。

适应度函数

路径规划的目标是找到一条从起点到终点适合机器人行走且较短的路径，所以目标函数的主要指标应该是评

价路径的质量，主要考虑以下几个方面：

- 路径长度，路径越短，机器人行走的时间就越短，则更适合机器人行走；
 - 路径上距离障碍物最近的距离，此距离越长，则机器人越不容易撞到障碍物，更安全；
 - 路径平滑度，平滑度越高，则机器人走起了越顺畅。
- 路径平滑度会在最后加上曲线优化，所以最后的适应度函数为：

		(4. 1)
--	--	--------

其中：

		(4. 2)
--	--	--------

a和b是用于控制路径长度的影响的参数，c为一个安全距离，表示路径上距离障碍物最近的距离小于c时有碰撞风险。计算路径到最近障碍物的距离是一个很耗时的操作，所以这里选择使用动态规划先计算好地图上每个点到最近障碍物的距离，计算路径时只需遍历一遍路径点选取最小值即可。

如何进行动态规划：把障碍物作为源点集、可行点作为目标点集求最短路，则对于每个可行点(x, y)，离它最近的障碍物点到它的路径只有四种情况，分别为从上下左右四个方向得到。考虑做两次动态规划，第一次从左上到右下，第二次从右下到左上，记f(x, y)为(x, y)距离最近的可行点的曼哈顿距离，则可以推出这样的转移方程：

第一次：

		(4. 3)
--	--	--------

第二次：

		(4. 4)
--	--	--------

初始化的时候把障碍物点的 f 值全部预置为 0，可行点的 f 全部预置为 INF（无穷大），做完两个阶段的动态规划后即可得到每个点距离最近的障碍物点的距离。

经测试，在此地图中a=10000, b=5, c=10是一个比较好的结果。即

		(4. 5)
--	--	--------

- 遗传算子
- 选择算子

这里不使用轮盘赌方法进行选择，而是使用单一保留方法，即只保留适应度最高的那个个体复制N次当作父代来生成子代群体，因为主要是对一条最优的路径进行优化。

交叉算子

因为要使得交叉后的路径仍然是可用的路径，所以不能直接使用第三章中的随机交叉的方法，这里路径交叉的方法为：

- 随机选择两个待交叉的路径，生存0-1之间的随机数是否小于交叉概率，若小于则进行交叉操作；
- 寻找两个路径是否有交叉点；
- 若有，随机选择一个非起始点的交叉点进行交叉；
- 若无，分别随机选择两条路径上的非起始点的一个点，在这两个点之间使用4. 2路径初始化中的方法生成一条路径，然后进行交叉操作，示例：假设待交叉的两条路径分别为：，随机选择的两个点为，在两点之间生成了路径，则交叉之后的路径为：

$path_1 = \{p_{11}, p_{12}, p_{13}, \dots, p_{1l_1}\}, path_2 = \{p_{21}, p_{22}, p_{23}, \dots, p_{2l_2}\}$

		(4. 6)
--	--	--------

且此操作在选择算子之前执行，因为选择算子中采用单一保留的方法，若在选择操作之后执行，则会因为父代完全一样而没有效果。

变异算子

同样的，变异操作也不能使用随机变为等位基因这种方法，因为并没有使用二进制编码，且可能导致产生间断路径，而此问题需要保证路径的可用性，所以首先想到使用随机删除点进行修补：

原路径，按照概率判断是否路径需要变异操作；

$$path=\{p_1,p_2,p_3,\dots,p_l\}$$

若需要，随机删除路径中非起始点的一个点；

$$p_r$$

将删除点的前后点作为起始点使用路径生成算法生成一条路径；

$$p_{r-1},p_r',\dots,p_{r+1}$$

将生成的路径拼接到原路径即完成了变异操作：

$$path'=\{p_1,p_2,p_3,p_{r-1},p_r',\dots,p_{r+1},\dots,p_l\}$$

但是这时会发现这种变异方式有一些问题，因为是按照路径初始化的方式来修补路径的，且步长较大，有很大的可能性，变异后的路径和原路径相同，即删除点的前后点作为起始点生成的路径仍经过删除点，或变异后的路径种群中本身就存在，这样就起不到变异操作产生新的优秀基因的作用了，无法增加种群多样性，不利用产生最优解。所以本文提出一种寻找周围可用点的变异方式：

首先仍是概率判断点是否变异；

假设要变异的点为，寻找此点距离为随机数的可用邻居点，若没有则缩短距离继续寻找；

$$p_r$$

在邻居点中随机选择一个判断之间是否有障碍物，若无则使用替换完成变异操作，若有则重新选择，直至完成变异操作或迭代完邻居点无法完成变异；

$$p_r'$$

使用此方法可有效的产生新的优秀基因，增加了种群的多样性。

曲线优化

完成遗传操作后，因为大部分都是步长不为1的移动，生成的最优路径是由折线连成的路径，所以最后需要将折线路径进行曲线化。

三次Bezier曲线由四个点确定，在两点之间绘制，需要两个控制点，要使得曲线的接头处顺滑，使得曲线函数的一阶导数连续即可，也就要使得曲线的切线斜率是连续的。假设起始点分别为（为 $()$ ），要确定的控制点为，过作曲线的切线，使得前面的控制点和后面的控制点，都经过这个切线，把切线的方向取为的方向。那么控制点的坐标就是：

$$P_i,P_{i+1}$$

		(4. 7)
--	--	--------

其中a, b为任意给定的正数，决定了曲线的弯曲的程度，经实验，是一组较为理想的取值，同时上式子用到的不存在的点使用替代。

$$a=b=\frac{1}{8}$$

直接使用经过的点作为控制点也是一个方法，但是需要路径点的个数为，所以可以在前个点使用直接 bezier，然后在剩下的点中再插入控制点。两种方法对比如图 4. 5所示。

$$3*n+1, n \in N^*$$

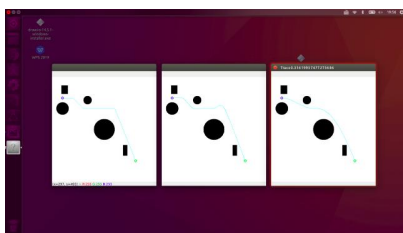


图 4.5 两种曲线化方法对比图

第一种方法经过了每个原本路径的点，但是会产生一些不需要的弯曲，第二种方法的效果更好。

结果分析

使用上述方法，使用参数：初始种群大小200，交叉概率0.4，变异概率0.1进行了1000次迭代实验，得出数据如表 4.2，和适应度值随着迭代次数的变化如图 4.6所示。

表 4.2 不同迭代次数情况

迭代次数	最大适应值	最短路径长度	时间/s
初始	20.7	503.55	2.60
迭代100次	22.8	493.34	6.48
迭代400次	24.2	486.45	20.01
迭代600次	24.5	480.37	29.36
迭代800次	24.5	480.37	38.70

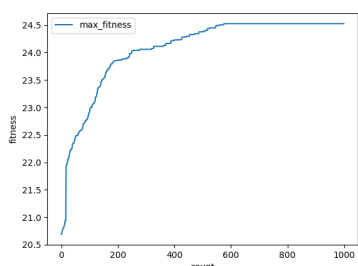


图 4.6 迭代次数与适应度值图

由迭代次数与适应度值图 4.6可见算法初始化之后就具有了很好的基因，刚开始的路径适应度就已经很高了，遗传算法过程中主要对优秀的基因进行了保留，并优化了与障碍物的最短距离，大概在600次迭代时已经基本收敛，经过曲线优化最后得出一条可用的优秀路径。

算法的其中一个主要耗时操作在于初始化路径中的计算每个栅格点最近的障碍物的距离，这个步骤对于同一副地图来说只会执行一次，所以计算每个点距离最近障碍物的距离不是算法的瓶颈，由于python的执行效率较低，在将算法移植到c++之中之后，效率可以进一步得到提升，经测试大约可以提升100倍左右，所以基本可以做到秒级响应。最后得到的未经曲线优化的路径如图 4.7所示，曲线优化后的如图 4.8所示。

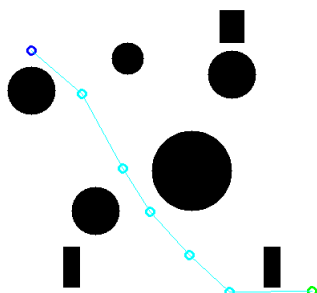


图 4.7 未曲线优化前的路径图

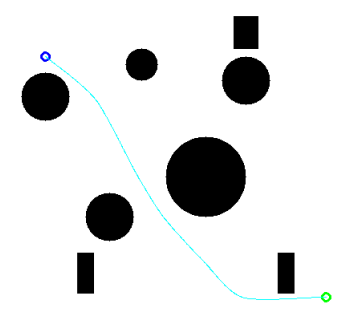


图 4.8 曲线优化后的最终路径图

路径规划仿真实验

章节标识符（隐藏）

仿真环境搭建

如第二章中所介绍，本实验主要在ROS系统中进行。所以首先是在Ubuntu系统中安装ROS Kinetic，直接使用 `sudo apt-get install ros-kinetic-desktop-full` 安装桌面完整版ROS。然后对ROS进行初始化：`sudo rosdep init && rosdep update`。

Gazebo作为ROS常用的模拟工具，在桌面完整版ROS中会被直接安装，所以无需再去手动安装Gazebo。如第二章中所介绍本实验使用中科院软件所提供的一个软件技术博物馆模型（如图 5.1）作为场景和XBot-U机器人（如图 5.2）作为实验对象。下载这两个模型文件后使用 `roslaunch` 启动对应的 `launch` 文件在Gazebo中载入世界场景和机器人模型后就可以看到具体搭建好的场景与机器人。

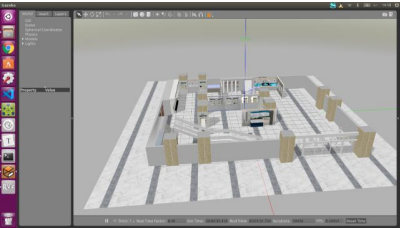


图 5.1 软件技术博物馆模型图

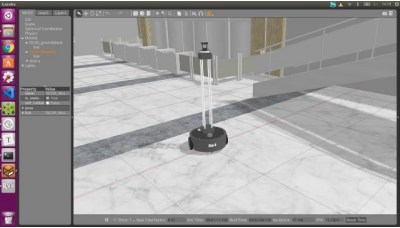


图 5.2 机器人XBot-U

除了Gezebo之外，还需要用到 `rviz` 这个软件，这是一个机器人可视化工具，可以用于查看构建地图或路径规划的具体情况，它也会在完整版ROS中直接安装。

在ROS系统环境中，一个机器人是通过多个节点一起运行并互相通信来工作的。**`roslaunch`可以帮助按照所编写的 `launch` 文件中的规则批量启动多个节点**，减少依次手动启动节点的麻烦，并且 `roslaunch` 会首先检测 `master` 是否正在运行，**`master`是ROS提供的一个节点管理器，在整个ROS系统通信网络里相当于管理中心**，管理着每个节点，当节点启动时，它将首先在节点管理器中注册，然后节点管理器将该节点合并到整个ROS程序结构中，所以也免去了手动 `roscore` 启动 `master` 的步骤。

具体的环境参数如表 5.1 仿真实验环境参数表所示。

表 5.1 仿真实验环境参数表

描述	值
----	---

系统	Ubuntu 16.04
机器人框架	ROS Kinetic 1.12.17
仿真工具	Gazebo 7.0.0
可视化工具	rviz 1.12.17
场景	软件技术博物馆
机器人	XBot-U
SLAM算法	GMapping SLAM
定位算法	AMCL

然后就是将上一章中所介绍的基于遗传算法的路径规划添加到ROS系统中以供ROS使用。因为路径规划是一个比较看中时间消耗的功能，所以这里使用性能更好的C++作为开发语言。

要使用自定义全局路径规划算法，需要遵守navcore包中定义的navcore::BaseGlobalPlanner接口以插件的形式注册到ROS当中，以便它可以被move_base包使用。所以按照以下步骤将算法注册到ROS中：

使用catkin_create_pkg命令在工作空间中创建一个新的package以编写路径规划类；

创建一个继承nav_core::BaseGlobalPlanner类的新的路径规划类，并重写其中的“void initialize(std::string name, costmap_2d::Costmap2DROS* costmap_ros)”和“bool makePlan(const geometry_msgs::PoseStamped& start, const geometry_msgs::PoseStamped& goal, std::vector< geometry_msgs::PoseStamped> & plan)”两个方法，这两个方法的作用分别是初始化路径规划类和做具体的路径规划。

在路径规划类实现文件中包含<pluginlib/class_list_macros.h>头文件，并使用PLUGINLIB_EXPORT_CLASS宏注册此插件；

在CMakeLists.txt文件中添加add_library以使得catkin_make可以将这个类编译为插件，再执行catkin_make进行编译；

添加插件描述文件以描述插件的一些信息以及允许ROS系统自动的发现，加载生成的插件；

在package.xml添加export部分将插件注册到ROS包系统。

最后就可以使用“rospack plugins --attrib=plugin nav_core | grep 插件名称”命令查找到注册好的插件。

完成插件注册后就可以具体去实现路径规划中的方法了，每次更改完代码之后只需再执行一次catkin_make即可让代码生效。

在initialize方法中先初始化一些要用到的参数，ROS系统提供了一个权重地图，这个地图是不会改变的，所以先在这个初始化方法中将权重地图构建为一个栅格地图类，并计算每个栅格点距离最近障碍物的距离以供在适应度函数中使用，在之后的路径规划过程中则不再需要每次计算这个距离。

然后就是实现makePlan方法，此方法传入要做路径规划的起点和终点，传出一条路径，但是要注意到一些坐标需要转换，此接口传入的坐标并不是相对于栅格地图的坐标而是实际世界的坐标，所以需要作如下转换：

		(5.1)
--	--	-------

其中，resolution为地图分辨率，即一像素等于多少米，(origin_x,origin_y)为地图原点，round为四舍五入函数。同样的，当生成路径之后，需要反向将路径中的坐标还原为实际世界坐标返回给ROS，ROS会帮助通过路径获得机器人下一步的两轮速度，所以无需关注前进和转向。

另外创建一个遗传法路径规划类来做路径规划算法，在上述的makePlan函数中实例化这个类，并将栅格地图与转换后的起始点坐标传入，计算完成之后再取得最后的路径进行返回，其它均直接使用C++移植自上一章中的python算法。

实验设计

如第二章中介绍，先使用GMapping Slam算法构建地图，在rviz中就可以查看具体情况，如图 5.3 使用Gmapping SLAM构建地图所示，构建好的地图如图 5.4所示，然后在路径规划时使用AMCL算法提供机器人的定位，在启动文件中指定启动amcl节点即可。

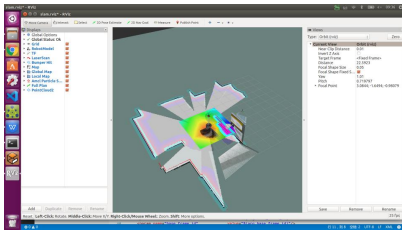


图 5.3 使用Gmapping SLAM构建地图



图 5.4 仿真环境中使用Gmapping生成的地图

将编写好的插件注册到ROS之中后，就可以依次启动ROS中的节点，使用rqt_graph命令可以查看到节点之间通信情况如图 5.5所示。

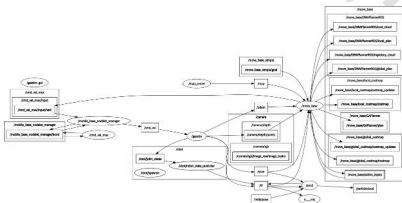


图 5.5 在ROS中的节点通信图

按照4.6节所介绍设置参数，迭代次数设置为500，使用dwa_local_planner作为局部路径规划算法。启动所有节点之后使用rviz设置机器人在地图上的目标姿态进行路径规划，如图 5.6所示。此处进行多次在不同的位置设置不同的起点与终点进行测试，包含多种不同的路径长度并进行记录。

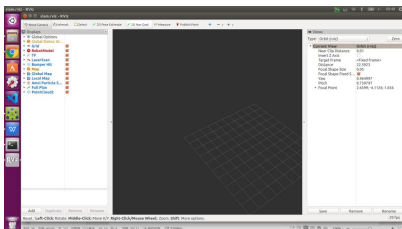


图 5.6 使用rviz设置机器人目标姿态

同时也可以从rviz中查看路径如图 5.7所示。

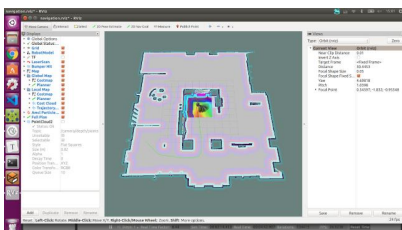


图 5.7 在ROS中使用遗传算法进行全局路径规划图

其中绿色线为规划所得的路径，实体墙的边缘处气泡为ROS计算了机器人的大小之后给出的虚拟墙。

然后对相同的起始点使用本算法与ROS系统Navigation Stack这个软件元包中其他的全局路径规划算法进行路径规划并记录路径长度与时间等信息。

实验结果分析

进行多次路径规划测试后，得到的数据如表 5.2所示。

表 5.2 不同起始点路径规划结果表

两点距离 (m)	路径长度 (m)	时间 (ms)
2.45	2.45	532
4.42	4.43	644
4.45	4.47	632
6.67	7.71	841
7.79	8.32	876
9.15	13.64	1021
11.26	18.12	983

相同起始点（起始点（4.95，0.00），（0.42，-7.48）距离约为8.74m）多个路径规划算法对比如表 5.3所示：

表 5.3 相同起始点不同路径规划算法效果对比

使用的路径规划算法	路径长度 (m)	时间 (ms)
parrot_planner	-	262
navfn	9.32	347
global_planner	NO PATH!	NO PATH!
本文的算法	9.35	852

使用parrot_planner、navfn与本文算法对相同起始点做路径规划生成的路径图分别如图 5.8、图 5.9、图 5.10所示。

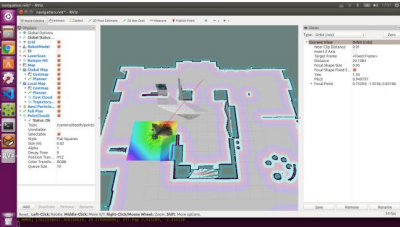


图 5.8 使用parrot_planner用作路径规划

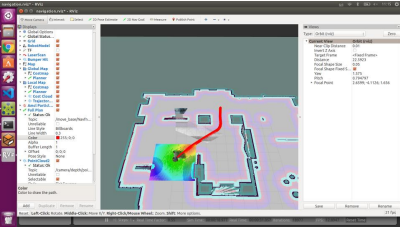


图 5.9 使用navfn用作路径规划

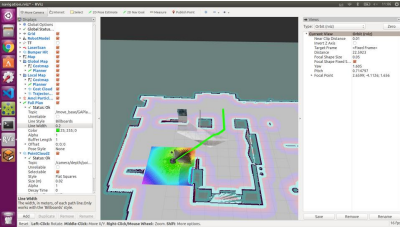


图 5.10 使用本文的算法用作路径规划

得益于C++的高性能，虽然速度没有ROS中自带的navfn中的迪杰斯特拉算法快，但是基本也可以做到秒级响应，且算法的适应能力较强，在多种情况下都可以在一个比较平均的时间内得到路径。parrot_planner由于只是寻找下一个距离目标点近的新目标点而未能得到可用的路径，global_planner也因为A*算法自身的缺陷出现NO PATH。本文生成的路径长度与使用navfn得到的路径相当，通过路径规划仿真实验，使用遗传算法可以保证机器人安全的到达目标点，证明了遗传算法在路径规划中的可行性。

总结与展望

本文的主要研究工作和结论：

阐明了基本遗传算法机制，介绍了遗传算法的基本原理与流程与几个基本的遗传算子以及如何去实现这些遗

传算子和一些优化方法。

基于遗传算法提出一种新的全局路径规划流程，研究了遗传算法在路径规划中的使用，通过比较几种不同的初始化路径的方法，逐渐优化了初始化路径的方法，然后对初始化的路径选择一条最优路径使用遗传算法优化，最后再进行了曲线优化。

将路径规划遗传算法作为插件移植到ROS中，在ROS环境中使用Gazebo进行了机器人仿真，介绍了仿真实验的平台，使用基于粒子滤波的Gmapping Slam算法构建栅格地图，使用AMCL提供机器人的位置，并进行了机器人坐标与栅格地图坐标的相互转换，使用rviz来查看地图路线以及设置目标点，通过多次在不同的位置设置起始点使用遗传算法与曲线优化对机器人进行路径规划，结果表明此算法在机器人全局路径规划中是可行的。

本文虽然完成了遗传算法在机器人全局路径规划中的研究、实现和在ROS中仿真，但仍然存在一些不足，后续可能还需优化的工作有：

仿真环境中各种数据都是最优的，在实际环境中会存在各种不确定性，所以考虑使用真实的机器人在实际环境中进行实验；

将曲线优化剥离在了遗传算法之外，虽然加入了安全距离但是仍然有小概率可能导致优化后的路径穿过障碍物，在ROS系统中可以依靠局部路径规划来解决，但是通用性不强，所以可以将曲线优化结合到遗传算子之中。

致谢

时光荏苒，岁月如梭，不知不觉，四年的大学时光就要过完了。

首先，我要感谢我的毕设指导老师杨林权副教授。从选题到论文完成，杨老师都给予了细致的指导，让我学习到了很多关于机器人路径规划的知识。

然后，我要感谢母校地大，感谢母校的老师、工作人员。正是学校有这么多好的老师、负责任的工作人员，才能把学校塑造成艰苦朴素、求真务实的形象。

我还要感谢与我朝夕相处的同学们，感谢他们对我生活上的关心和学习上的帮助。

最后，我要感谢我的父母、亲人们。正是我最爱的人们在背后的默默付出，支撑着我继续砥砺前行，迎接我的研究生生活。

衷心祝愿我身边的每个人都能够健康快乐，幸福平安。

参考文献

- Qu L, Wang H. An overview of Robot SLAM problem[C]//2011 International Conference on Consumer Electronics, Communications and Networks (CECNet). IEEE, 2011: 1953-1956.
- Dijkstra E W. A note on two problems in connexion with graphs[J]. Numerische mathematik, 1959, 1(1): 269-271.
- Gang Peng, Wei Zheng, Zezao Lu, Jinhua Liao, Lu Hu, Gongyue Zhang, Dingxin He, Junpei Zhong. An Improved AMCL Algorithm Based on Laser Scanning Match in a Complex and Unstructured Environment[J]. Complexity, 2018, 2018.
- Khatib O. Realtime Abstract Avoidance for Manipulators, and Mobile Robots in Proe [J]. IEEE Int Conf. On. Robotics and Automation March 25 -381985. 500 -505, also in Int JRobot Res, 1986, 5(1):90-98.
- 王奇志. 基于改进人工势场法的多障碍机器人运动控制 [C]. 北京: 2003 年中国智能自动化会议论文论文集(上册)
- Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing[J]. science, 1983, 220(4598): 671-680.
- 袁曾. 人工神经网络及其应用[M]. 北京:清华大学出版社, 1999, 10.
- Holland J H. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence[M]. MIT press, 1992.
- 焦合军, 周万春, 李渊博. 基于混合遗传算法的机器人路径规划研究[J]. 中州大学学报, 2020, 37(06): 125-128.
- 胡章芳, 程亮, 张杰, 王春瑞. 多约束条件下基于改进遗传算法的移动机器人路径规划[J/OL]. 重庆邮电大学学报(自然科学版): 1-8[2021-05-04]. <http://kns.cnki.net/kcms/detail/50.1181.n.20210316.1753.042.html>.
- Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1994, 24(4): 656-667.

翁理国, 王安, 夏旻, 纪壮壮. 基于改进SPEA2算法的机器人路径规划研究[J]. 计算机仿真, 2014, 31 (07) :346-350.

王雷, 李明, 唐敦兵, 等. 基于改进遗传算法的机器人动态路径规划[J]. 南京航空航天大学学报, 2016, 48 (6) : 841-846.

杨嘉, 刘虎, 杨新坤, 李文振, 李富康, 赵宁宁. 基于遗传算法的移动机器人路径规划[J]. 机电工程技术, 2020, 49 (12) :97-98+117.

Robotics; Reports on Robotics Findings from Xuzhou Institute of Technology Provide New Insights (Path Planning of Mobile Robot Based On Hybrid Multi-objective Bare Bones Particle Swarm Optimization With Differential Evolution) [J]. Journal of Robotics & Machine Learning, 2019.

Changfu Yang, Tao Zhang, Xihao Pan, Mengyang Hu. Multi-objective mobile robot path planning algorithm based on adaptive genetic algorithm[A]. 中国自动化学会控制理论专业委员会 (Technical Committee on Control Theory, Chinese Association of Automation)、中国自动化学会 (Chinese Association of Automation)、中国系统工程学会 (Systems Engineering Society of China). 第三十八届中国控制会议论文集 (3) [C]. 中国自动化学会控制理论专业委员会 (Technical Committee on Control Theory, Chinese Association of Automation)、中国自动化学会 (Chinese Association of Automation)、中国系统工程学会 (Systems Engineering Society of China):中国自动化学会控制理论专业委员会, 2019:7.

刘洋, 马建伟, 臧绍飞, 闵义博. 基于融合Bezier优化遗传算法的路径规划[J]. 控制工程, 2021, 28 (02) :284-292.

Jianwei Ma, Yang Liu, Shaofei Zang, Lin Wang, Raşit Köker. Robot Path Planning Based on Genetic Algorithm Fused with Continuous Bezier Optimization[J]. Computational Intelligence and Neuroscience, 2020, 2020.

宋宇, 王志明. 基于改进遗传算法的移动机器人路径规划[J]. 现代电子技术, 2019, 42 (24) :172-175.

段立伟, 王进华. 改进遗传算法在移动机器人路径规划中的应用[J]. 信息通信, 2020 (07) :5-8.

魏彤, 龙琛. 基于改进遗传算法的移动机器人路径规划[J]. 北京航空航天大学学报, 2020, 46 (04) :703-711.

李国洪, 王远亮. 基于B样条和改进遗传算法的机器人时间最优轨迹规划[J]. 计算机应用与软件, 2020, 37 (11) :215-223+279.

Hao Kun, Zhao Jiale, Yu Kaicheng, Li Cheng, Wang Chuanqi. Path Planning of Mobile Robots Based on a Multi-Population Migration Genetic Algorithm. [J]. Sensors (Basel, Switzerland), 2020, 20 (20).

Grisetti G, Stachniss C, Burgard W. Improved techniques for grid mapping with rao-blackwellized particle filters[J]. IEEE transactions on Robotics, 2007, 23 (1) : 34-46.

Charles Darwin's theory of evolution: An analysis[J]. Journal of the History of Biology, 1975, 8 (2).

Theodosius D. Genetics and the Origin of Species[J]. 1937.

Holland J.H. Outline for a logical theory of adaptive systems [J]. Journal of the Association for Computing Machinery, 1962, 9 (3) :297-314.

霍凤财, 迟金, 黄梓健, 任璐, 孙勤江, 陈建玲. 移动机器人路径规划算法综述[J]. 吉林大学学报(信息科学版), 2018, 36 (06) :639-647.

De Jong K A. An analysis of the behavior of a class of genetic systems[J]. Dissertation Abstracts International, 1975, 36: 5140B.

Grefenstette J J. Optimization of control parameters for genetic algorithms[J]. IEEE Transactions on systems, man, and cybernetics, 1986, 16 (1) : 122-128.

Schaffer J D, Caruana R, Eshelman L J, et al. A study of control parameters affecting online performance of genetic algorithms for function optimization[C]//Proceedings of the 3rd international conference on genetic algorithms. 1989: 51-60.

指标说明:

1、总相似比: 类似于重合率, 即送检论文内容与所选检测资源范围内所有文献相似的部分 (包括参考引用部分)

），占整个送检论文内容的比重，总相似比=复写率+引用率；

2、复写率：即送检论文内容与所选检测资源范围内所有文献相似的部分（不包括参考引用部分），占整个送检论文内容的比重；

3、引用率：即送检论文内容中被系统识别为引用的部分，占整个送检论文内容的比重（引用部分一般指正确标示引用的部分）；

4、自写率：即送检论文内容中剔除相似片段和引用片段后，占整个送检论文内容的比重，一般可用于论文的原创性和新颖性评价，自写率=1-复写率-引用率；

5、同届相似比：即送检论文内容与校方所选同届库检测资源范围内所有文章相似的部分（不包括参考引用部分），占整个送检论文内容的比重；

6、报告中，**红色**与**橙色**文字表示复写片段，**蓝色**与**深蓝色**文字表示引用片段，**紫色**文字表示同届相似片段，黑色文字表示自写片段。

免责声明：

1、本报告为G·格子达系统检测后自动生成，鉴于论文检测技术及论文检测样本库的局限性，G·格子达不保证检测报告的绝对准确，仅对您所选择的检测资源范围内的检验结果负责，相关结论仅供参考，不做法律依据；

2、G·格子达论文检测服务中使用的论文样本，除特别声明者外，其著作权归各自权利人享有。根据中华人民共和国著作权法相关规定，G·格子达网站为学习研究、介绍、评论、教学、科研等目的引用其论文片段属于合理使用。除非经原作者许可，请勿超出合理使用范围使用其内容和本网提供的检测报告。

联系我们：



防伪二维码



关注微信公众号

官方网站: www.gocheck.cn

客服热线: 400-699-3389

客服QQ: 800113999