



廣東財經大學  
GUANGDONG UNIVERSITY OF FINANCE & ECONOMICS

广东财 经大学		2025-JX16-

# 本科毕业论文

## 基于遗传算法的校园路径规划研究

学院	信息学院
专业	软件工程
班级	2 班
学号	21251104256
学生姓名	袁智泉
指导教师	周雅兰
提交日期	2025 年 4 月 28 日

# 广东财经大学本科生毕业论文（设计） 原创性声明

本人郑重声明：所呈交的论文（设计）是本人在指导老师指导下独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文研究做出过重要贡献的个人和集体，均已在文中以明确的方式标明。因本论文（设计）引起的法律后果完全由本人承担。

作者签名： 袁智泉

2025 年 4 月 15 日


指导老师签名： 周雅兰

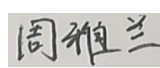
2025 年 4 月 15 日

# 本科毕业论文（设计）版权使用授权书

本学位论文作者完全了解广东财经大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的纸质版和电子版，允许论文被查阅和借阅。本人授权广东财经大学可以将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、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名： 

指导老师签名： 

签字日期：2025 年 4 月 15 日    签字日期：2025 年 4 月 15 日

毕业论文题目： 基于遗传算法的校园路径规划研究

软件工程 专业 21 级

本科生姓名： 袁智泉

## 摘要

随着互联网和智能手机的普及，智能路径规划服务取得了迅速的发展，人们如何对校园内各个地点从单一最短路径转向多属性综合优化是当前校园出行路径规划的一大难题。文章利用基于遗传算法的多属性路径规划方法，选取路径长度、树荫覆盖、路灯照明与景观优美程度四个关键因子，构建兼顾效率和环境舒适度的个性化路径选择模型。文章利用图论模型建立校园网络结构模型，使用 MATLAB 软件对路径做了仿真实验。实验结果表明，文章中算法能够有效根据个性化需求建立差异化路径，能够提升路径规划的适应性和实用性，具有一定的应用价值，能为校园智能出行提供理论支持和技术参考。

**关键词：**多属性路径规划；遗传算法；图论建模；动态权重调整

TITLE: Research on Campus Path Planning Based on Genetic Algorithms

MAJOR: software engineering

APPLICANT: Yuan Zhiquan

## **Abstract**

With the popularity of the Internet and smart phones, the intelligent path planning service has made rapid development. How to turn the single shortest path to multi-attribute comprehensive optimization for each location in the campus is a major problem in the current campus travel path planning. Using the multi-attribute path planning method based on genetic algorithm, this paper selects four key factors: path length, shade coverage, street lighting and landscape beauty, and constructs a personalized path selection model that takes into account efficiency and environmental comfort. This paper uses graph theory model to establish the campus network structure model, and uses MATLAB software to do the simulation experiment on the path. The experimental results show that the algorithm in this paper can effectively establish differentiated paths according to personalized needs, improve the adaptability and practicability of path planning, and has certain application value, which can provide theoretical support and technical reference for campus intelligent travel.

**Keywords:** Multi-attribute path planning; Genetic algorithm; Graph theory modeling; Dynamic weight adjustment

# 目 录

<b>1 绪论</b>	1
1.1 研究背景与意义	1
1.2 研究现状分析	1
1.3 研究目的与内容	2
1.3.1 研究目的与研究问题	3
1.3.2 研究内容与创新点	3
<b>2 相关理论与技术基础</b>	4
2.1 经典的路径规划算法	4
2.1.1 Dijkstra 算法与 A*算法	4
2.1.2 传统算法的局限性与改进方向	5
2.2 遗传算法基本原理与优化策略	5
2.2.1 遗传算法的基本流程	6
2.2.2 遗传算法的选择、交叉与变异策略	7
2.2.3 遗传算法的特点	8
<b>3 校园路径规划问题建模设计</b>	9
3.1 问题描述	9
3.2 路网的表示	10
<b>4 基于遗传算法的校园路径规划设计</b>	12
4.1 实验环境	12
4.2 遗传编码	12
4.3 初始种群的选取	12
4.4 适应度函数的构造	12
4.4.1 四种因素的权重系数计算	13
4.4.2 四种因素量纲统一	14
4.4.3 适应度函数计算	15

4.5 选择操作设计 .....	15
4.6 交叉操作设计 .....	16
4.7 变异操作设计 .....	16
<b>5 算法仿真实验结果与分析 .....</b>	<b>17</b>
5.1 实验结果 .....	17
5.2 结果比对与分析 .....	18
5.2.1 与传统 Dijkstra 算法的对比分析 .....	18
5.2.2 基于遗传算法的方法结果分析 .....	20
<b>6 结论与展望 .....</b>	<b>21</b>
6.1 研究成果总结 .....	21
6.2 研究的应用前景 .....	21
<b>参考文献 .....</b>	<b>22</b>
<b>附录一 .....</b>	<b>24</b>
<b>附录二 .....</b>	<b>26</b>
<b>附录三 .....</b>	<b>30</b>
<b>附录四 .....</b>	<b>32</b>
<b>致谢</b>	

# 1 绪论

## 1.1 研究背景与意义

随着城市化进程的加快和城市交通压力的增加，路径规划问题也逐渐成为一个日益重要的研究课题<sup>[1][2]</sup>。诸如 Dijkstra 算法和 A\*算法等经典的路径规划算法在求解最短路径问题方面表现优异，但是现代人工智能技术和计算机技术的飞速发展，使得人们不再满足于原有的简单环境下的路径规划，期望能有一种在动态障碍物、高维空间或实时性要求严格的高复杂度环境下，均能实现个性化的路径选择，提高用户的个性化维度。

校园路径规划是一个典型的多属性路径规划问题。校园用户在选择自己的步行路径时，会关心最短路程，会考虑如树荫、路灯、景观等环境因素<sup>[3]</sup>。这些因素在不同时间段对于不同需求的校园用户有着差异化的权值比重，本研究会在提升路径最优性与环境适应性等多个优化目标之间进行有效权衡。

遗传算法是一种模拟生物进化过程的智能优化方法，该算法凭借出色的全局寻优能力，在多目标与高维度的复杂优化问题中表现出显著优势且在路径规划及相关领域获得了广泛应用。本研究将遗传算法与多属性路径规划进行融合，实现动态调整不同因素的权重的同时，会进一步提高路径选择的合理性与优化程度<sup>[4][5]</sup>。

本研究旨在提出一种基于遗传算法的校园多属性路径规划方法，该方法突破传统算法在多属性优化方面的局限性并为校园内的导航系统与智能交通系统等提供更加精准和更为高效的路径规划方案。

## 1.2 研究现状分析

在智能交通、机器人导航、物流等领域，路径规划的研究得到了广泛关注。经典路径规划算法（包括 Dijkstra 算法、A\*算法以及 Bellman-Ford 算法等）在最短路径求解领域已形成成熟的理论体系，并在各类路径探索问题中展现出不错的应用效果<sup>[6][7]</sup>。多属性路径规划问题是在最优路径规划的问题上拓展开来的，最优路径问题是一个比较经典的且值得探索的问题，放在当今，是运筹学图论、地理科学、人工智能、计算机科学等多领域的研究热点。目前，最优路径规划已经从单约束的单一交通工具和单一影响因素发展成为综合的统筹的多种交通工具和多种影响因素的多目标路径规划模型。随着多种环境因素的引入，传统方法的局限性也逐渐暴露出来<sup>[8][9]</sup>。



国内外的研究主要集中在以下两个方面，考虑了如时间、费用与舒适度等多个约束条件的优化型多目标路径规划算法和基于遗传算法与蚁群算法等通过全局搜索找到最优解的智能优化算法的应用。随着多因素和多目标优化问题的引入而导致的问题复杂度的提升，经典的路径规划算法已经难以应对复杂多变的环境。

针对多属性路径规划需求，Google Maps 服务提出了采用多种动态因素相结合，进行实时路径优化。随着研究的不断深入，路径选择问题开始由单目标向多目标转化。Google Maps 不仅考虑了交通流量、施工、天气等多种实时信息来提供最优路径选择，而且针对用户设置的如避免高速公路收费和避免交通堵塞等偏好来实现多属性路径规划的同时向用户提供个性化的路线推荐[10]。

当前国内学术界在该领域的研究主要集中于静态权重模型的构建，对于动态权重调整机制的研究仍较为匮乏，在多因素实时交互影响下的自适应权重优化方面尚待深入探索。国外现有的研究大多集中在大范围城市交通系统中，而针对小范围复杂环境的路径规划模型仍需要进一步深入研究。也就是说，国外在针对多属性路径规划问题的研究和针对校园等特定环境的路径规划研究是相对较少的。

本研究将针对多属性校园路径规划问题展开细致的研究，对遗传算法提出了实现策略，用来求解问题，在提高算法的运行的合理性的同时提升最终结果的可行性，并可为将来路径规划服务提供一定的理论指导和技术参考。

## 1.3 研究目的与内容

### 1.3.1 研究目的

本研究的主要目的是针对校园环境中的多属性路径规划问题，提出一种基于遗传算法的优化模型，探讨如何通过动态权重调整来提高路径规划的灵活性和适应性。随着校园中路径中用户个性化的客观存在性，越来越多的人需要一个可以权衡多需求的校园步行路径规划服务来辅佐人类进行路径规划。本研究整合如路程、树荫、路灯、景观优美程度等多种因素，实现更符合用户需求的路径规划<sup>[11]</sup>。

### 1.3.2 研究内容与创新点

本研究分析路径网络的构成，构建校园路径网络模型，总结该图论模型的特征，将影响路径选择的主要因素（如路程、树荫、路灯、景观优美程度等）转化为权重指标，给出路径网络模型的评价标准并分析。本研究进而总结出多属性路径问题的类型，归纳最优路径分析算法的分类体系，设计一种改进的遗传算法，解决多属性

路径规划的全局优化问题。

本研究基于 MATLAB 仿真环境，实现所提出的适应的遗传算法，并对校园路径规划问题进行仿真实验。本研究在应用层次分析评价，归纳改进遗传算法在多维度优化中的优势与不足。本研究会对实验结果进行详细的讨论与总结。

本研究的创新点有以下几个方面：第一，算法引入动态权重调整机制。针对多属性路径规划问题，本研究提出了基于环境变化的动态权重调整方法，这个权重调整方法不同于传统静态加权模型，提高路径规划的适应性和灵活性；第二，研究融合遗传算法与多属性优化。本研究采用遗传算法解决多目标路径规划问题，提升算法的全局优化能力；第三，范围限定在校园路径规划特定场景。针对校园这一特定场景，本研究构建了符合实际需求的多属性路径规划模型，使路径规划更加符合用户的个性化需求。

## 2 相关理论与技术基础

本章主要介绍与本研究相关的理论与技术基础。本章详细阐述遗传算法的基本原理，对路径规划中常见的传统算法进行分析，提出将遗传算法应用于多属性路径规划问题的方案。本章讨论遗传算法的特点，以支持后续算法的设计和实现操作。

### 2.1 经典的路径规划算法

路径规划作为图论与人工智能领域的重要研究方向，在历史长河中，已发展出多种经典算法。例如，1959年，荷兰计算机科学家狄克斯特拉发表了具有创新性的有关提出 Dijkstra 算法的论文；1962年美国的计算机教授 Robert Floyd 基于动态规划的思想提出了一种全所未有的路径规划算法，后人则用教授的名字来命名此算法；1968年，三位科学家 B.Raphael, P.E. Hart 和 N.J. Nilsson 三人共同发表论文，提出了 A-Star 这一重要的路径搜索算法。它们都是代表性的经典的路径规划算法，现在常常被用于网络内部路由问题的求解或者作为其它的复杂图论算法的子算法辅助进行计算。本节将对 Dijkstra 算法与 A\*（A-Star）算法进行分析与比较，并讨论其在本研究中可以借鉴或待改进的地方。

#### 2.1.1 Dijkstra 算法与 A\*算法

##### 1. Dijkstra 算法

Dijkstra 算法是典型的单源最短路径计算算法，此算法可以解决由初始顶点到图中其他所有顶点之间的最短路径计算问题<sup>[12]</sup>。该算法采用的是贪心算法的算法思想，该算法会以起始点为中心，根据边的权重大小向外处理当前路径代价最小的节点，并且不断迭代地更新邻接节点的最短路径估计值，直至到达目标节点才停止。该算法适用于带权图且权重必须为非负数。从一般情况来看，该算法的时间复杂度为  $O(V^2)$ ，当采用优先队列对算法进行优化过后，优化后的时间复杂度可降到可观的  $O((V + E) * \log_2 E)$ ，其中  $V$  是图中结点数， $E$  是图中边数，且  $V \leq E$ 。

##### 2. A\*（A-Star）算法

A\*算法是 Dijkstra 算法的改进版，是一种常见的启发式算法，是一种用启发信息来进行路径搜索的算法<sup>[13]</sup>。A\*算法的核心是构建估值函数，通过构建的估值函数来对当前节点到终点进行代价估值，代价函数  $f(x) = g(n) + h(n)$ ，其中  $g(n)$  表示从起点到当前节点的实际代价， $h(n)$  为启发式估计函数，该函数用于预估从当前节点

n 到目标节点的最小代价。由于 A\* 算法通过引入这种启发式估计，在大多数情况下比 Dijkstra 算法搜索速度更快。其时间复杂度受启发式函数影响，通常介于  $O(N \log_2 N)$  和  $O(N^2)$  之间。

A\* 算法在如智能导航系统的实际工程领域中有着较为不错的实用性。若 A\* 算法采用的是恰当的启发式评估函数，该算法能有效缩小需探索的解空间范围，并大规模提升路径规划效率。常用的启发函数有曼哈顿距离与欧几里得距离等。

### 2.1.2 传统算法的局限性与改进方向

从以往文献上看，传统路径规划算法在特定场景中效果优秀，但在现代复杂场景背景下，会暴露出以下的一些不足：

第一，缺乏多属性处理能力。实际路径选择常涉及多个维度，如距离、安全性、景观、美观度、环境舒适度等多种有着可个性化性的动态可调节因素，传统算法通常仅支持单一权重的优化，难以满足用户多元化和个性化的偏好<sup>[14]</sup>；

第二，对环境变化响应相对的不足。传统算法较多关注静态图模型，针对动态的更新机制部分的研究比较薄弱，这种模式难以应对复杂多变的实时变化，例如突发施工、天气变换、道路拥堵等不可预测的变化；

第三，启发式设计依赖性较强。A\* 算法性能过度依赖于启发函数设计，算法的适用性会受限，导致难以自适应调整；

第四，传统算法的搜索空间大，效率提升空间少。在大规模图或稀疏图中，传统算法的效率相对来说并不是很高，较难满足用户的个性化要求。

面对复杂的多因素环境，我们的改进方向主要包括了以下几点。第一，实验引入如遗传算法、蚁群算法、粒子群优化等智能优化算法，并对路径多目标优化进行考虑；第二，算法构建动态权重模型，根据环境与用户偏好调整路径选择标准；第三，算法融合图论结构建模，提升搜索效率与解的多样性；第四，实验利用机器学习方法训练路径偏好预测模型，为路径评分提供强有力的数据支撑。

总的来看，本研究在多属性路径规划场景下将遗传算法与图论建模相结合，可以提升算法对多因素约束的适应能力，还能提高整体路径选择的灵活性与智能化水平，为传统路径规划算法提供重要补强与升级方向。

## 2.2 遗传算法基本原理与优化策略

仿生学一方面模仿生物界的现象和原理，而另外一些研究方向则是以遗传算法为代表的对自然现象或过程的模拟，算法将达尔文的生物进化论和孟德尔的遗传学说融入到算法中。遗传算法（Genetic Algorithm，GA）是模拟

生物在自然环境中的遗传和进化过程而形成的一种自适应全局优化概率搜索算法。该算法通过模拟种群的选择、交叉和变异等遗传能力，在组合优化、路径规划及人工生命等多个领域展现出强大的全局搜索能力<sup>[15]</sup>，是 21 世纪有关智能计算中的关键技术之一。

## 2.2.1 遗传算法的基本流程

遗传算法的基本流程大致可以被分为以下的几个步骤<sup>[16]</sup>：

初始化种群：这一步骤指的是第一代原始染色体群体的产生。每个个体采用染色体形式编码，编码方案可根据问题特性选择二进制串与实数向量等不同表示方法，编码中的每个染色体对应问题的一个可行解。

构建适应度函数：在研究自然界中生物的遗传和进化现象时，生物学家使用适应度这个术语来度量某个物种对于生存环境的适应程度。从更直观的角度来看，我们可以将适应度函数看作为衡量一个个体好坏的标准。在路径规划问题中，适应度函数需要综合考虑路径长度和用户个性化需求等多个优化指标，目的是为了全面评价解的优劣程度。

选择操作：这一步骤以一定的概率从种群中选择若干个体，并以适应度值筛选优质个体作为繁殖父本，通俗来说，就是从上一代种群中选取具有优秀基因的个体来进行复制。直观地看，我们参照适应度函数将较高适应度的个体赋予较大的概率，使它有更多的机会传递到下一代中去。选择操作的设计对算法的性能起着决定性作用。

交叉操作：有性生殖生物在繁殖下一代的同时，同源的染色体通过交叉而重组，在两个染色体的某一位置处 DNA 被切断，其前后两串分别交叉组合而形成两个新的染色体。该操作的核心机制是通过基因片段的交换，来实现解空间的探索。如何合理的制作交叉算子是遗传算法中比较棘手的问题。一方面，为了保持种群在进化过程中的多样性，它需要尽可能多的制造出不同的个体。另一方面，还要避免破坏优秀个体，并且加快算法的收敛速度。所以，要想使种群的多样性和收敛性达到高度统一，就应该考虑到以上两个方面，它们是衡量一个交叉算子好坏的标准。

变异操作：变异操作通过引入随机扰动维持种群基因多样性，有效防止算法过早收敛于局部最优。该操作以预设的小概率对染色体特定基因位进行随机调整，通常变异概率控制在 0.1%-1% 范围内。在一般的遗传算法中，对于生成新品种的能力来说，交叉算子是要大于变异算子的，因为它是整个算法中产生新的染色体的最主要的手段。

结束条件判断：算法终止机制通过预设的收敛准则进行触发。当实验满足任一终止条件时，进化过程就会立即终止，并输出当前种群中的最佳个体作为最终解。

主要流程如图 2-1 所示。

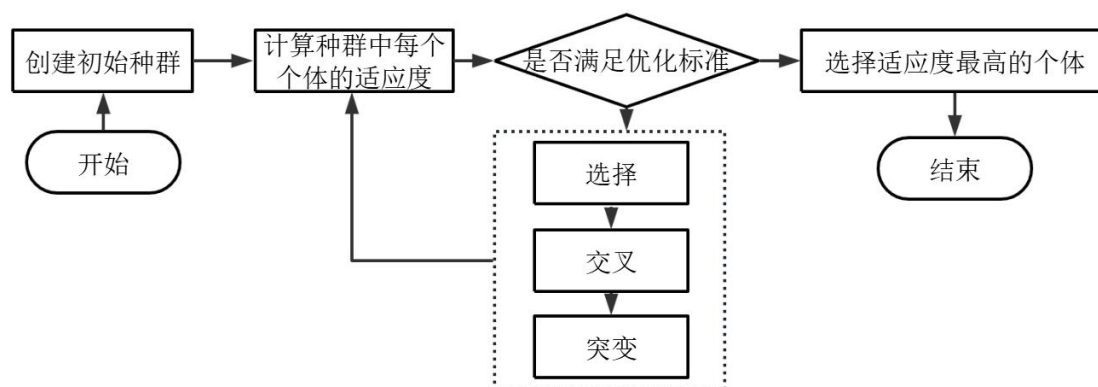


图 2-1 遗传算法流程图

Figure 2-1 Flow chart of genetic algorithm

## 2.2.2 遗传算法的选择、交叉与变异策略

选择、交叉和变异是遗传算法中三个至关重要的操作，它们共同影响着算法的优化效果和搜索能力。

**选择操作。**选择操作的目的是从当前种群中选出适应度较高的个体，作为下一代的父代。常见的选择策略有以下几种。第一，轮盘赌选择。轮盘赌选择基于个体适应度值在群体中的相对占比来分配选择概率，适应度越高的个体获得越大的选择机会。该方法通过概率映射的手段实现优胜劣汰，提升个体质量；第二，锦标赛选择。锦标赛选择从群体中随机抽取若干个体组成竞赛组，直接比较组内个体的适应度值并选择最优者。这种方法计算简单，但容易受到局部最优解的影响；第三，排名选择。排名选择对全体个体按适应度排序，根据预设的概率映射函数选择概率，排名越高的个体被选中的概率就会越大。

**交叉操作。**交叉操作的主要功能是通过父代个体间的基因交换来产生新的候选解。典型的基因重组方式包含以下几种。第一，单点交叉。单点交叉在染色体编码序列中随机确定一个交叉位点，互相交换两个父代个体在该位点前后的基因片段，这会生成两个新的后代个体。第二，两点交叉。两点交叉与单点交叉类似，随机选取两个不同的交叉位点，交换父代个体在这两个位点之间的基因序列。第三，均匀交叉。均匀交叉让每个基因位独立地以相同概率进行交换，随机重组基因。

**变异操作。**变异操作是对个体基因进行小幅度随机修改，避免算法陷入局部最优解的困境中。常见的变异方法有以下几种。第一，单点变异。单点变异在染色体序列中随机选取一个基因位置，按照预设规则对该位点的基因值进行修改。这种局部微调地方法有助于保持种群一定的稳定性。第二，多点变异。多点变异随机地选择多个不连续的基因位置进行协同修改。这种方式能够产生更明显的多样性，但需要控制变异强度。第三，位翻转变异。位翻转变异针对采用二进制编码方案的个体，

以特定概率反转基因位的逻辑状态（0 变为 1，1 变为 0），这种操作适合处理离散型优化问题。

实现通过合理地选择适应度函数和调整选择、交叉、变异操作的设计参数，使遗传算法在不同的优化问题中发挥更好的效果。

### 2.2.3 遗传算法的特点

遗传算法作为一种优化的启发式搜索方法，能归纳在群智能算法的一类算法中，遗传算法与大多数早早以前提出的传统的经典的搜索和优化算法有着很大的区别，遗传算法主要有着以下特点：

#### （1）并行性。

遗传算法从初始种群中开始搜索，单个种群中所有个体的初始化、选择、交叉和变异的操作具有相对独立性，可以从不同的方向搜索，使得算法内部具有较高的并行性。另外，可以将种群分布在多个进程，实现物理意义上多个种群相对独立地进行算法迭代，在得出最优解的过程中各个种群并不需要进行通信，只需要在算法迭代完成后，进行解的比较，从而得出最优解<sup>[17]</sup>。

#### （2）良好的通用性和鲁棒性。

传统的优化算法需要用一个算数表达式来表示所要解决的问题，而且还要要求该函数表达式的一阶二阶导函数是存在的。遗传算法在应用时不需要提前知道任何的辅佐信息，可以仅凭借适应度函数来评估控制个体的优劣程度。遗传算法对适应值的计算函数没有要求和限制，目标的好处不需要连续、可微等限制，甚至还不需要明显的函数形式，在算法过程中不需要其他的干预，这使得遗传算法不受外界因素变化的影响，其通用性比其他算法无可比拟的<sup>[18]</sup>。

总结来说，遗传算法使一种区别其他算法的全新算法，它主要进行全局的优化，该算法简单有效，并且使用，适合多任务处理。基于上述的特点，遗传算法可以广泛地运用在各个学科领域，如机器学习、深度学习、组合优化和路径规划等等。

### 3 校园路径规划问题建模设计

在路径规划问题中，如何对校园路网进行准确的建模的这个挑战，是实现路径优化的第一步，也是根基。本研究就通过对校园内道路、建筑物、绿地等环境因素的综合考虑，建立符合实际需求的网络图论模型。本章将介绍校园路径优化问题的建模方法，包括问题的描述和路网的表示。

#### 3.1 问题描述

本研究的主要问题包括：第一，多属性路径规划建模问题。建模问题的实质是如何有效地将如树荫、路灯等的不同属性转化为可量化的指标，并在路径规划中综合考虑这些因素；第二，权重调整问题。权重调整问题的实质是如何根据时间的变化对各属性的权重进行动态调整，以确保路径规划结果的合理性和适应性。第三，遗传算法的应用问题。应用问题的实质是如何设计有效的染色体编码方案、构建合理的适应度评价函数，以及优化选择、交叉和变异等遗传操作策略，确保算法在复杂约束条件下具备高效的全局寻优能力和收敛精度。主要环节见图 3-1。

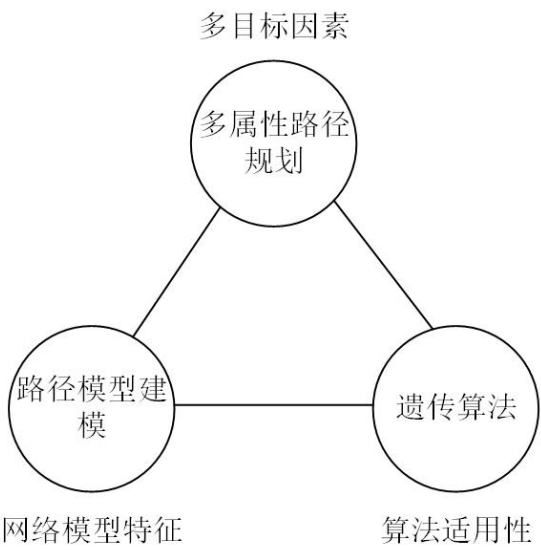


图 3-1 多属性最优路径分析组成

Figure 3-1 Composition of Multi-attribute Optimal Path Analysis

本研究把路径的多重影响因素考虑在内，合理地设计一个能够根据用户需求进行个性化选择的最优路径的模型，来解决校园的路径选择问题。我们把用户的需求设定为路径的最短距离、最佳观景体验、最大树荫覆盖度以及最优的照明条件。简单来说的话，本问题的目标是寻找一条能够在多个因素上达到综合最优效果的步行路径。



该问题的建模要考虑上述因素，并结合合理的权重分配，使得不同的用户可以根据自己的需求调整各个因素的优先级，得到一条符合个人需求的个性化自适应的最优路径。

## 3.2 路网的表示

我们可以把校园路径规划问题近似地看为图论中的最短路径问题，只不过本研究在多属性影响层面进行了考虑。校园内的每一条关键道路和每一座主要建筑物可以表示为图中的节点和边。这些节点之间有着相互的连接关系，如道路的属性 and 距离，这些属性可以表示为图中边的权重大小。本研究把校园道路网络抽象为带权无向图，这种抽象方法可以有利于遗传算法的计算处理，能有效刻画路径规划中的各类约束条件。

图中的节点代表校园中的一个如教学楼、宿舍楼、食堂和操场等的建筑物主要地点。节点的编号则根据校园地理位置或逻辑规划来进行定义，本研究通过空间关系，从上至下，从左至右进行编号。图中的边代表节点之间的有效连接路径，路径可以是道路、人行道或其他通道。

每条边都有一个自己的权重值，权重值通常可以表示为路径的代价，如距离、时间、舒适度、观景度等个性化因素。实际情况来看，本研究把图中的边权重值设定为以下几种情况：第一，距离因素。距离权重表示两个地点之间的实际路程，其单位设定为米。第二，树荫因素。树荫权重表示路径是否存在树荫覆盖。值为 1 则表示存在树荫覆盖，值为 0 则表示无树荫覆盖。这个权重主要考虑了路径的舒适度。第三，观景因素。观景权重表示路径的是否存在景观。值为 1 则表示存在景观，值为 0 则表示无景观。这个权重主要考虑了用户的观景需求。第四，路灯因素。路灯权重表示路径是否有路灯。值为 1 则表示存在路灯，值为 0 则表示无路灯。这个权重主要考虑了路径的安全性和夜间通行的明照程度。

本研究以广东省广州市某高校的校园道路布局作为仿真案例的对象，布局如图 3-2 所示。本研究设定对应的路径拓扑网络模型如图 3-3 所示。

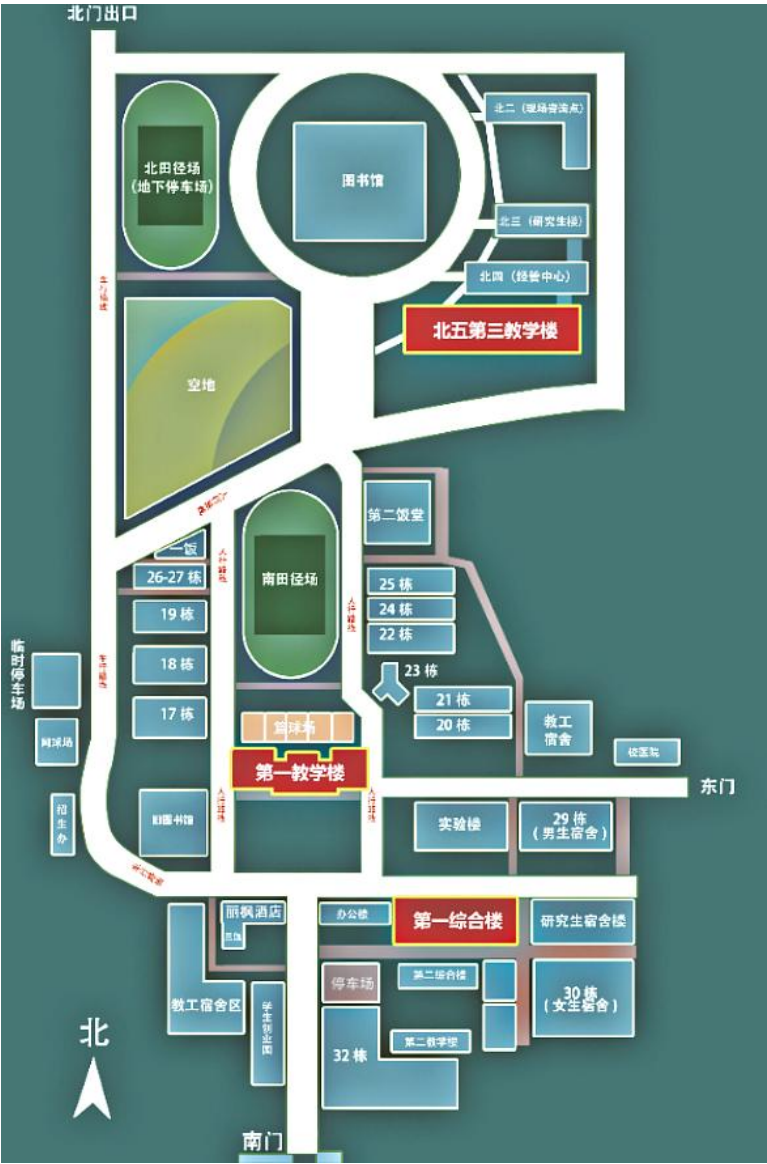


图 3-2 广东省广州市某高校地图

Figure 3-2 Map of a University in Guangzhou, Guangdong Province

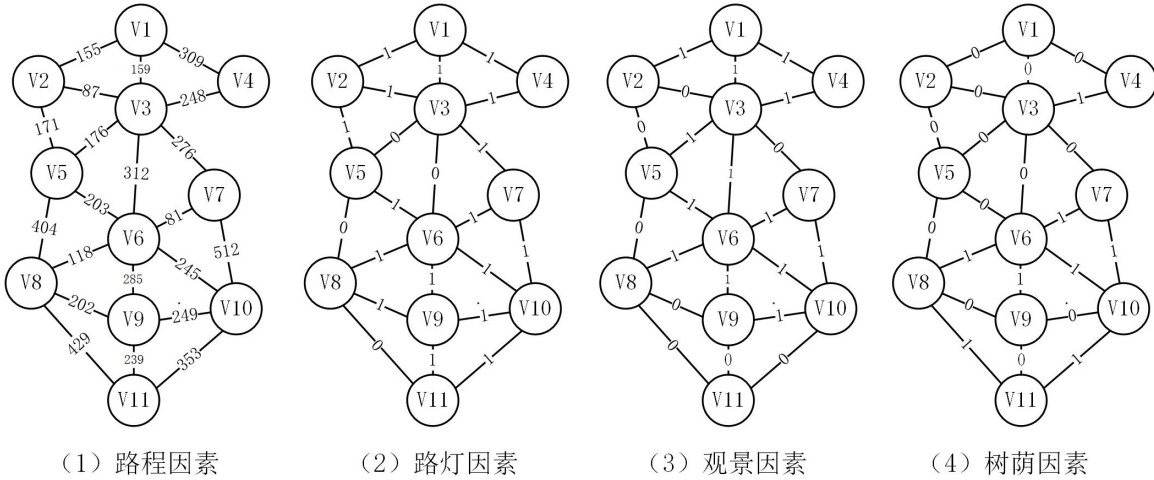


图 3-3 道路网络拓扑结构示意图

Figure 3-3 Schematic Diagram of Road Network Topology Structure

该拓扑网络模型图是一幅无向带权图，其节点表示的是校园内的关键地点，其边表示连接这些节点的可通行路段，其权重对应地表示距离、树荫、观景和路灯因素。本研究的数据来源于高德地图提供的校园地理信息。本研究通过高德地图软件提取了校园内主要建筑物（如教学楼、宿舍楼、食堂、操场等）的位置坐标，并根据实际路径连接关系构建了节点和边<sup>[19]</sup>。以下则为路程因素的邻接矩阵图可视化图，如图 3-3 所示。

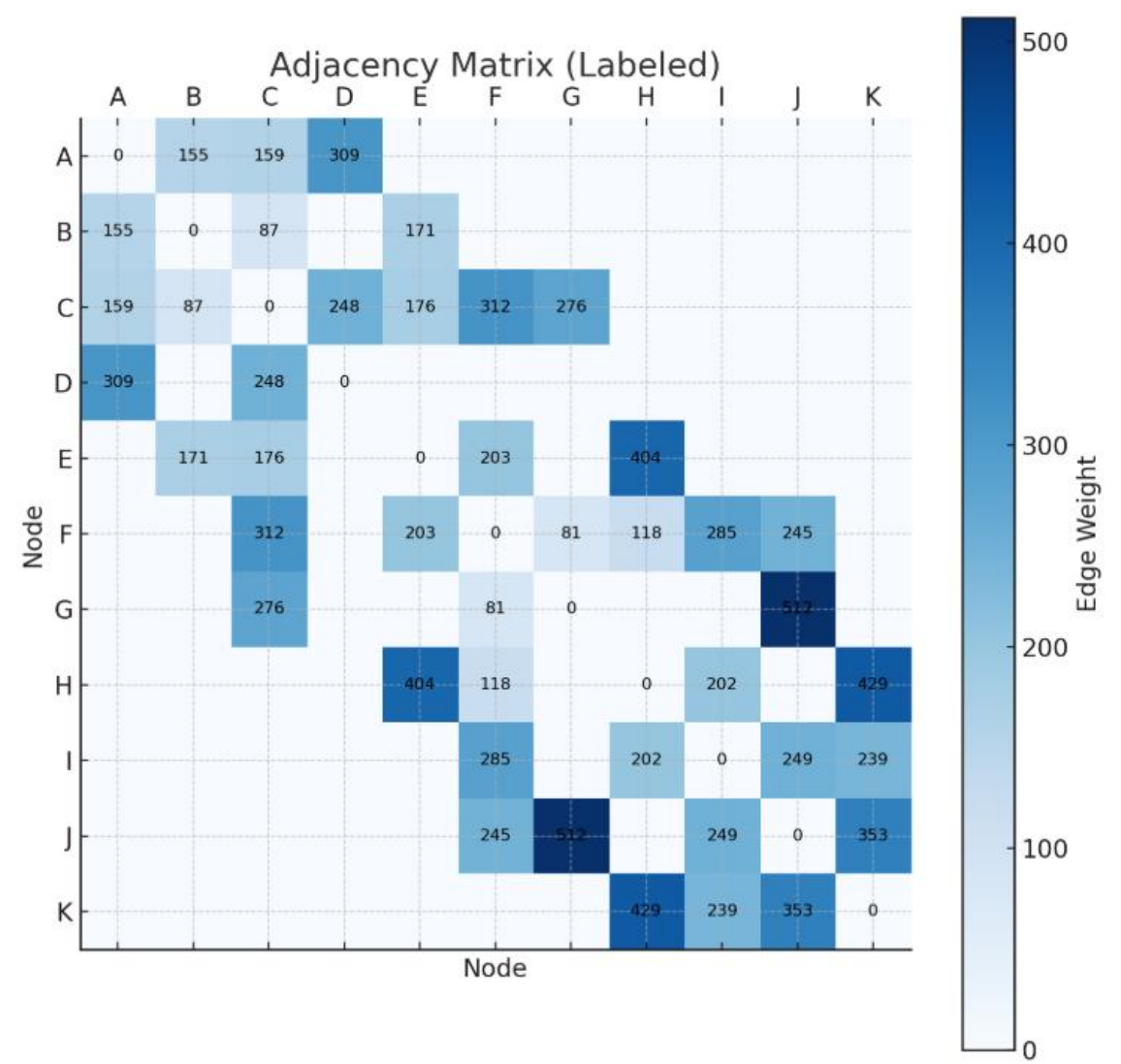


图 3-3 邻接矩阵图可视化图

Figure 3-3 Visualization of Adjacency Matrix Graph

图中，同一节点间的加权距离为 0，无通路的两节点之间的距离不显示，编程时用相当大的值  $\text{inf}$  代替。以下为关键实验代码：

```
% 距离矩阵（无连接设为 inf）
A = [0 155 159 309 inf inf inf inf inf inf inf;
155 0 87 inf 171 inf inf inf inf inf inf;
159 87 0 248 176 312 276 inf inf inf inf;
inf inf inf inf inf inf inf inf inf inf inf;
inf inf inf inf inf inf inf inf inf inf inf;
inf inf inf inf inf inf inf inf inf inf inf;
inf inf inf inf inf inf inf inf inf inf inf;
inf inf inf inf inf inf inf inf inf inf inf;
inf inf inf inf inf inf inf inf inf inf inf;
inf inf inf inf inf inf inf inf inf inf inf;
inf inf inf inf inf inf inf inf inf inf inf];
```

```
309 inf 248 0 inf inf inf inf inf inf;
inf 171 176 inf 0 203 inf 404 inf inf inf;
inf inf 312 inf 203 0 81 118 285 245 inf;
inf inf 276 inf inf 81 0 inf inf 512 inf;
inf inf inf inf 404 118 inf 0 202 inf 429;
inf inf inf inf inf 285 inf 202 0 249 239;
inf inf inf inf inf 245 512 inf 249 0 353;
inf inf inf inf inf inf inf 429 239 353 0];
```

```
A = min(A, A'); % 确保对称
```

## 4 基于遗传算法的校园路径规划设计

本章实验采用遗传算法实现校园路径规划，实验的主要流程包括编码设计、种群初始化、适应度评估和进化操作四个阶段。首先，我们将校园路径抽象为染色体，采用十进制编码表示节点序列，如此确保路径的连续性和有效性。初始种群来随机生成一组满足连通性约束的可行解。随后，我们构建多目标适应度函数，综合考虑路径长度、树荫覆盖、景观质量和夜间照明等因素，用层次分析法确定各指标权重。在进化阶段采用混合选择策略，我们结合精英保留和轮盘赌选择，平衡收敛速度与种群多样性。交叉操作通过交换父代路径的公共节点片段生成新路径，变异操作则基于邻域替换调整局部节点，均设置约束条件保证路径有效性。算法迭代执行选择、交叉和变异操作，一系列操作直至满足终止条件并最终输出最优路径方案。我们的整个流程在 Matlab 仿真环境下完成，这样可以确保算法可复现性和结果可靠性。

### 4.1 实验环境

AMD R7 5800H 八核 CPU，8G 内存，操作系统为 Windows 11，用 Matlab 仿真实现改进的遗传算法。

### 4.2 遗传编码

染色体编码（或称个体编码）在路径优化问题中起着关键作用，其设计质量直接影响到遗传算法中选择、交叉与变异操作的效果。在本研究中，将一条路径作为一个染色体，其中路径上的每个节点对应一个基因位点<sup>[20]</sup>。为简化个体表示，本算法采用固定长度的编码方式，例如，广东省广州市某高校道路结构中中共有 11 个结点，则染色体的长度为 11。本算法拟采用十进制编码。采用十进制编码取代二进制编码来直接表征参数，其好处是取消了编码、译码过程，有助于提升算法运行效率和数值精度并有效规避了由于基因串过长而带来的运算复杂性。此外，路径设计中引入了约束条件：从起点至终点的路径中不得重复经过同一节点，且不允许将没有直接连接关系的节点连续排列。

### 4.3 初始种群的选取

初始种群是随机选择的一组有效候选解(个体)。由于遗传算法使用染色体代表每

个个体，因此初始种群实际上是一组染色体。

4.4 适应度函数的构造

在遗传算法中，适应度函数（亦称评价函数）用于衡量个体在种群中的优劣水平，是指导进化方向的核心指标<sup>[21]</sup>。本研究从路径选择的多因素特性出发，综合引入路程长度、树荫覆盖程度、夜间照明条件（路灯）及景观美观度四项关键因素。为合理确定各因素在用户决策中的相对重要性，采用层次分析法（Analytic Hierarchy Process, AHP）对个体用户的偏好进行建模，计算出相应的权重系数，从而增强模型对个性化需求的适应能力<sup>[22]</sup>。

4.4.1 四种因素的权重系数计算

为实现校园路径规划中多重因素的综合权衡（包括路程、树荫覆盖、景观质量及照明条件等），本研究引入了层次分析法（Analytic Hierarchy Process, AHP）以确定各影响因素的权重系数。AHP 的基本原理是通过构建成对比较的判断矩阵，并计算其特征向量，从而量化各因素在决策过程中的相对重要性<sup>[23]</sup>。在本节中，我们将详细介绍权重系数的计算过程，具体包括以下三个步骤：

第一，构建判断矩阵。结合路径规划常识与用户行为逻辑，依据四个因素在不同时段对用户决策影响的差异，构建了四个因素之间的相对重要性判断矩阵。采用了 1-9 标度法，矩阵中的每个元素表示一个因素相对于另一个因素的重要性。具体的判断矩阵如表 4-1，与表 4-2 所示。其中，元素 $a_{ij}$ 表示第  $i$  个因素与第  $j$  个因素的相对重要性，数值参考 Saaty 标度法（1 - 9 标度），并依据路径使用场景合理设定。

表 4-1 白天判断矩阵  
Table 4-1 daytime judgment matrix

	路程	树荫	观景	路灯
路程	1	2	5	2
树荫	1/2	1	4	2
观景	1/5	1/4	1	1/2
路灯	1/2	1/2	2	1

表 4-2 夜晚判断矩阵  
Table 4-2 Night Judgment Matrix

	路程	树荫	观景	路灯
路程	1	5	3	5

树荫	1/5	1	3/5	1
观景	1/3	5/3	1	5/3
路灯	1/5	1	3/5	1

第二，计算特征根和特征向量。完成判断矩阵的构建后，我们要进一步求解其特征根及对应的特征向量，用来提取各因素的权重信息。然后通过计算判断矩阵的特征向量，来得到每个因素的相对权重。

用和法<sup>[20]</sup>计算白天路程影响因素权重：

$$\begin{aligned}w_1 &= \frac{1}{4} \times \left( \frac{1}{1 + \frac{1}{2} + \frac{1}{5} + \frac{1}{2}} + \frac{2}{2 + 1 + \frac{1}{4} + \frac{1}{2}} + \frac{5}{5 + 3 + 1 + 2} + \frac{2}{2 + 2 + \frac{1}{2} + 1} \right) \\&= 0.392\end{aligned}$$

同理可得其余三种影响因素权重分别为：0.298、0.133、0.177，如表 4-3 所示。同理，夜晚影响因素权重如表 4-4 所示。

表 4-3 白天影响因素权重  
Table 4-3 Weight of daytime influencing factors

因素	路程	树荫	观景	路灯
权重	0.392	0.298	0.133	0.177

表 4-4 夜晚影响因素权重  
Table 4-4 Weight of Influencing Factors at Night

因素	路程	树荫	观景	路灯
权重	0.577	0.115	0.115	0.192

第三，一致性检验。

$$\begin{aligned}\lambda_{max} &= \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n d_{ij} w_j / w_i = \frac{1}{4} \sum_{i=1}^4 \sum_{j=1}^4 d_{ij} w_j / w_i = 4.0002 \\C_I &= \frac{\lambda_{max} - n}{n - 1} = \frac{4.0002 - 4}{4 - 1} = 0.000067 \\C_R &= \frac{C_I}{R_I} = \frac{0.000067}{0.90} = 0.000074 < 0.10\end{aligned}$$

以夜晚影响因素为例，上式中， $\lambda_{max}$ 为  $n$  阶判断矩阵的最大特征值； $n$  为影响因素个数； $d_{ij}$ 为判断矩阵中因素  $i$  对因素  $j$  的比重； $w_i$ 、 $w_j$ 分别为影响因素  $j$ 、 $i$  的权重； $C_I$ 为一致性指标； $C_R$ 为检验系数； $R_I$ 为随机一致性指标，判断矩阵阶数为 4 时取值为 0.90。同理可得，白天的检验系数为 0.048。因为  $C_R$  均小于 0.1，说明判断矩阵通

过一致性检验，可接受。

#### 4.4.2 四种因素量纲统一

在本多属性路径规划问题中，我们考虑的四个影响因素有路程、树荫、路灯和景观。这些因素的单位 and 量纲各自不一样，我们在构建适应度函数之前，必须对它们进行量纲统一，确保它们的影响权重可以公平地进行比较。

本研究中，设定的四个影响因素的量纲分别为：

- (1) 路程：单位为米（m）；
- (2) 树荫：二元值，表示路径上是否存在树荫（0 或 1）；
- (3) 路灯：二元值，表示路径上是否存在路灯（0 或 1）；
- (4) 景观：二元值，表示路径上是否存在景观（0 或 1）。

由于树荫、路灯和景观均为二元值（0 或 1），它们的值可以直接作为影响因素的数值，无需进行标准化处理。这些二元值已经具备了量纲一致性，实验可以直接用于计算。唯一需要标准化的是路程这一非二元值因素，因为它的单位为米，且值的范围可能较大，路程因素需要通过标准化将其转化为无量纲的数值，这样可以更为方便地让路程因素与其他因素进行合理的比较。

我们为了使路程与其他三个因素具备可比性，本研究采用最大值标准化法，将路程值转换为一个无量纲的标准化值。具体的标准化公式为：

$$D_{norm} = 1 - \frac{\sum_{k=1}^{n-1} A(path_k, path_{k+1})}{D_{max} - D_{min}}$$

其中， $A$  为距离矩阵， $path_k$  为路径第  $k$  个节点， $D_{min}$ 、 $D_{max}$  为路网中所有有效路径的最小和最大距离，路程的值会变成一个介于 0 和 1 之间的无量纲效益型指标：值越大表示距离越短。

树荫、路灯和景观：由于这三个因素的值为 0 或 1，分别表示是否存在树荫、路灯和景观，我们直接使用原始数值，将它们的二元值作为权重计算的一部分。具体数值可以在后续的加权计算中与路程的标准化值进行结合计算。

#### 4.4.3 适应度函数的构造

本研究的适应度函数综合结合了路径的路程、景观、夜间照明和日间遮荫四个指标，用加权求和实现多目标优化，并通过线性加权将多指标转化为单一适应度值，主要实现代码参考附录一。具体的适应度函数为：

$$F = w_1 \times D_{norm} + w_2 \times S_{norm} + w_3 \times L_{norm} + w_4 \times T_{norm}$$

其中， $F$  为适应度函数； $w_1$  为路程因素权重， $w_2$  为观景因素权重， $w_3$  为路灯因素权



重， $w_4$ 为树荫因素权重； $S$ 、 $L$ 、 $T$ 分别为观景、路灯、树荫的权值。

## 4.5 选择操作设计

本研究在个体选择阶段融合了精英保留机制与轮盘赌法，这样的选择操作设计兼顾收敛速度与种群多样性<sup>[24]</sup>。以下为选择操作的关键代码：

```
% 精英选择
eliteCount = round(Popsize * eliteRatio);
[~, eliteIndices] = maxk(scores, eliteCount);
new_population = population(eliteIndices);
% 轮盘赌选择
valid_indices = find(valid_flags);
if ~isempty(valid_indices)
    valid_scores = scores(valid_flags);
    valid_scores = valid_scores - min(valid_scores) + eps;
    prob = valid_scores / sum(valid_scores);
    cum_prob = cumsum(prob);
    for i = (eliteCount+1):Popsize
        r = rand();
        sel = find(cum_prob >= r, 1);
        if isempty(sel)
            sel = length(valid_indices);
        end
        new_population{i} = population{valid_indices(sel)};
    end
end
population = new_population;
```

这段 MATLAB 代码实现了遗传算法中的精英保留与轮盘赌选择相结合的混合选择策略。首先通过精英选择保留当前种群中适应度最高的前 eliteCount 个个体（eliteRatio 指定比例），然后使用轮盘赌选择从剩余个体中按适应度比例进行概率选择以补充种群。轮盘赌选择过程先对有效个体（valid\_flags 标记）的适应度值进行归一化处理，计算累积概率分布，再通过随机数  $r$  确定选择位置。该策略既保证了最优个体不被淘汰，又维持了种群多样性，其中精英选择比例和适应度标度转换方式（加 eps 避免零概率）是影响算法性能的关键设计。最终新种群由精英个体和轮

盘赌选择个体共同组成，完成一代选择操作。

具体一点来说，本研究的选择操作将当前种群中适应度值排名前 10% 的个体直接保留至下一代，让高质量解延续，这步骤有利于避免优良遗传信息的丢失。对于剩余部分个体，本研究依据轮盘赌策略进行概率选择，根据个体适应度值占总体适应度的比例，确定其被选中的概率，对潜在优解的进行了有效引导。具体的概率计算函数为：

$$p_i = \frac{f_i}{\sum_{i=1}^n f_i}$$

其中  $p_i$  为概率， $f_i$  为个体的适应度值。为提升选择效率，算法会预先排除所有无效路径个体。轮盘赌选择过程中，算法对有效个体的适应度进行归一化处理，得到选择概率。算法再通过累积概率分布和随机数生成确定被选个体。这种混合选择策略既保证了种群中最优解的保留，又通过概率选择维持了种群的多样性，使算法能够有效平衡局部搜索和全局探索能力。

## 4.6 交叉操作设计

本研究使用改进型顺序交叉策略进行交叉操作，利用父代路径中的共同拥有的节点作为交叉点，算法进行交换实现路径片段的重组。以下是交叉操作的关键代码。

```
% 交叉操作
for i = 1:2:Popsiz-1
    if rand() < Pc
        [population{i}, population{i+1}] = ...
        crossoverPaths(population{i},
            population{i+1}, A, B, C, D, start, goal);
    end
end
```

该 MATLAB 代码实现遗传算法的交叉操作，采用相邻配对策略遍历种群个体，以预设交叉概率  $Pc$  控制是否执行交叉。代码使用步长为 2 的循环结构依次选取相邻个体作为父代，通过 `rand()` 函数生成随机数与  $Pc$  比较决定是否触发交叉。当满足交叉条件时调用 `crossoverPaths` 函数执行具体交叉运算，该函数接收两个父代个体及路径规划相关参数（包括约束条件 A、B、C、D 和起止点 `start`、`goal`），确保生成的子代路径满足问题约束。交叉操作直接修改原种群数组，实现种群更新。这种设计既保持了遗传算法通过重组优良基因提升解质量的特性，又针对路径规划问题的特殊约

束进行了适应性调整，是算法实现种群进化的关键步骤。参数  $P_c$  的合理设置直接影响算法探索与开发能力的平衡，需要根据具体问题进行调整优化。

本研究的交叉操作先识别两个父代个体中相同的中间节点集合，从相同的中间节点集合中随机选取一个作为此轮交叉操作的交叉位置；随后以该节点为界，互换目标个体的后续路径片段，生成具备部分父代特征的子代个体。若想要确保路径有效性，子代必须需要满足所有相邻节点必须实际连通，且完整保留起点和终点。交叉后的两个子代染色体可能会存在回路，而且有可能存在多条回路,因为回路的存在，交叉后的子代染色体节点数也有可能超过原地图的节点总数,本例中有可能超过 11，导致种群矩阵列数增加,因此必须删除所有回路。若子代不满足条件，则算法放弃本次交叉操作并保留父代。上述交叉过程如图 4-1 所示。

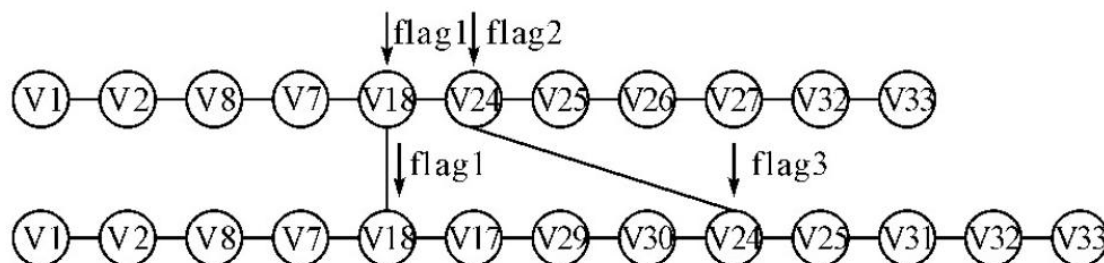


图 4-1 交叉过程

Figure 4-1 Cross Process

这种设计通过公共节点约束保证路径连续性，可以在维持种群多样性的同时有效传递父代优良特征<sup>[26]</sup>。交叉概率设为 0.8，原因是平衡探索与开发。

## 4.7 变异操作设计

本研究采用基于邻域替换的自适应变异操作。以下是变异操作的关键代码。

```
for i = 1:Popsiz % 变异操作
    if rand() < Pm
        population{i} = mutatePath(population{i}, A, start, goal);
    end
end
```

这段 MATLAB 代码实现了遗传算法的变异操作，遍历种群中的每个个体，以预设变异概率  $P_m$  决定是否执行变异。当满足变异条件时，调用 `mutatePath` 函数对当前路径进行随机修改，同时考虑路径约束  $A$  和起止点(`start,goal`)的限制。该操作通过引入随机性增强种群多样性，帮助算法跳出局部最优解，是维持遗传算法全局搜索能力

的关键机制。变异概率  $P_m$  的设置需要权衡探索与开发能力，通常取较小值以保证算法收敛性。

算法会先随机选择当前路径中的一个非端点节点作为变异位点，然后从距离矩阵中筛选出与该节点前后相邻节点直接连通的候选节点集合。然后算法随机替换候选节点实现局部路径调整，这样变异操作设计可以严格保证变异后路径的连通性和完整性。若新路径不满足路径有效性的约束条件，算法则放弃本次变异操作。算法把变异概率设置为 0.15，这个概率有利于平衡算法的探索能力与稳定性，该数值可以有效地在保持种群多样性的同时避免过度随机化。当路径长度处于较短的情况时，算法会自动降低变异强度以防止路径结构被过度破坏。这种变异机制在测试中展现出良好的适应性并维持了有效变异成功率。

5 算法仿真实现结果与分析

5.1 实验结果

本章主要基于第四章提出的多属性路径规划改进遗传算法模型，结合 MATLAB 实现进行仿真实验，并展示从校园图中任意两点间进行路径规划所得到的最优路径和综合得分。实验分别在“白天模式”和“夜晚模式”下进行，以体现不同时间权重设定对路径结果的影响。

实验中设置节点总数为 11，代表校园中 11 个主要位置。各节点间的连接情况通过带权邻接矩阵表示，考虑四个属性包括：

- （1）距离（距离越小越优）；
- （2）景观（景观丰富性，越高越优）；
- （3）路灯（夜间照明条件，越多越优）；
- （4）树荫（白天舒适性，越多越优）。

算法主要参数设置如下：种群规模为 300，最大迭代次数为 100，交叉概率为 0.8，变异概率为 0.15，精英保留比例为 0.1。以下是主要参数的关键代码：

```
%% 参数设置
pointnumber = 11; % 节点数量
Popsize = 300; % 种群规模
MaxGeneration = 150; % 最大迭代次数
Pc = 0.8; % 交叉概率
Pm = 0.15; % 变异概率
eliteRatio = 0.1; % 精英保留比例
```

以下为部分节点对路径的实验结果，由表 5-1 所示，完整实验结果请参考附录二。实验结果中，综合得分是该算法的一个关键指标，若一条路径的综合得分越高，表示该路径在综合考虑所有因素后的“优越性”越高，也就是说越合适用户的偏好。

表 5-1 遗传算法的部分节点对路径的实验结果

Table 5-1 Experimental Results of Partial Node-to-Path of Genetic Algorithm

起点	终点	白天最优路径	白天综合得分	夜晚最优路径	夜晚综合得分
1	11	1、3、6、8、11	0.9881	1、3、6、9、11	0.7147
1	5	1、3、5	1.4750	1、2、5	1.4591
4	11	4、3、6、8、11	0.9522	4、3、6、9、11	0.6431
6	11	6、10、11	1.3432	6、9、11	1.2935

8	1	8、6、3、1	1.3534	8、5、2、1	0.9207
---	---	---------	--------	---------	--------

此外，通过使用力导向布局实时显示最优路径演化过程，并在运行结束后用红色高亮显示所有有效路径，图中未用红色高亮标红的路径为无效路径，提供了更直观的动态可视化分析，如图 5-1 所示。

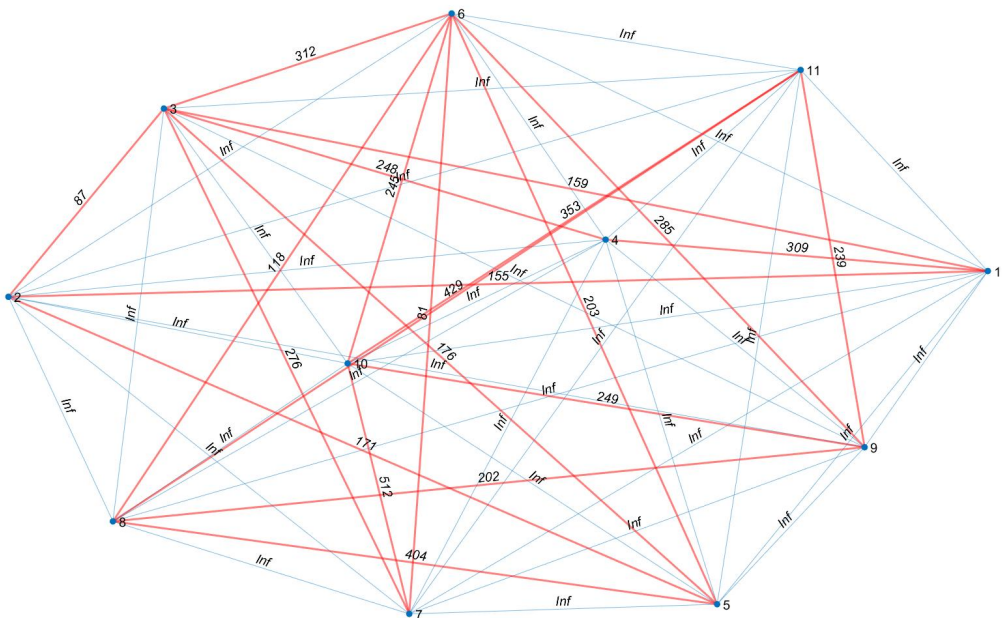


图 5-1 有效路径集合  
Figure 5-1 Effective Path Set

5.2 结果比对与分析

5.2.1 与传统 Dijkstra 算法的对比分析

本研究与传统求单源最优路径的 Dijkstra 算法作比较，主要实现代码参考附录三。实验数据采用上例中的 11 个节点的广东省广州市某高校的校园道路路径同条件下进行 Matlab 仿真实验。白天与夜晚模式的部分实验结果如表 5-2 所示，以下为 Dijkstra 算法的部分关键代码，完整实验结果请参考附录四。

```
current = -1;
min_d = inf;
for v = 1:n
    if ~visited(v) && distance(v) < min_d
        min_d = distance(v);
        current = v;
    end
end
end
```

```

if current == -1 || current == goal
    break;
end

```

以上代码是一个循环操作，操作选择当前未访问且距离最小的节点（标准 Dijkstra 逻辑），其中提前终止条件是到达目标节点或无可访问节点，其中的 distance 仍作为首要选择标准，保证路径可行性。

```

new_dist = distance(current) + A(current, neighbor);
norm_dist = 1 - (new_dist - min_dist)/(max_dist - min_dist);           % 归一化距离
norm_scenery = (B(current, neighbor) + B(neighbor, current))/2;        % 景观评分
norm_light = (C(current, neighbor) + C(neighbor, current))/2;          % 照明评分
norm_shade = (D(current, neighbor) + D(neighbor, current))/2;          % 树荫评分
new_score = weights(1)*norm_dist + weights(2)*norm_scenery + ...
            weights(3)*norm_light + weights(4)*norm_shade;             % 加权总分

```

以上代码是实现更新邻结点的多目标评分功能，计算路径的四个归一化指标：距离、景观、照明、树荫。通过权重系数 weights 实现多目标加权求和。其中，norm\_dist 的归一化公式  $(1 - (x - \min) / (\max - \min))$  将距离转化为效益型指标，双向评分取平均（如  $(B(\text{current}, \text{neighbor}) + B(\text{neighbor}, \text{current})) / 2$ ）增强鲁棒性。

```

if isinf(distance(goal))
    optimal_path = [start];           % 无解情况
else
    u = goal;
    while u ~= 0
        path = [u path];             % 反向追踪路径
        u = previous(u);
    end
    optimal_path = path;
    optimal_score = score(goal);      % 返回终点得分
end

```

以上代码实现的路径重构功能，通过 previous 数组反向重构最优路径，返回路径节点序列和终点综合得分。

实验结果中，综合得分是该算法的一个关键指标，若一条路径的综合得分越高，表示该路径在综合考虑所有因素后的“优越性”越高，也就是说越合适用户的偏好。

表 5-2 Dijkstra 算法的部分节点对路径的实验结果

Table 5-2 Experimental results of paths of some node pairs in Dijkstra algorithm

起 点	终 点	白天最优路径	白天综合得 分	夜晚最优路径	夜晚综合得 分
1	11	1、3、7、6、10、 11	0.7991	1、3、7、6、10、 11	0.6286
1	5	1、3、5	1.4335	1、2、5	1.4016
4	11	4、3、7、6、10、 11	0.7310	4、3、7、6、10、 11	0.5283
6	11	6、10、11	1.1942	6、10、11	1.2101
8	1	8、6、3、1	1.3220	8、6、7、3、1	1.1695

此外，我们根据属性满足能力对两种方法做出算法性能对比。我们采用的对比指标分别为路灯保障率（夜间场景）与树荫覆盖率（白天场景），具体公式如下式所示：

$$R_{light} = \frac{\text{路径中包含路灯的段数}}{\text{路径总段数}}$$

$$R_{shade} = \frac{\text{路径中包含树荫的段数}}{\text{路径总段数}}$$

上式中， $R_{light}$ 表示路灯保障率，用于量化路径沿线照明设施的覆盖程度， $R_{shade}$ 表示树荫覆盖率，反映路径的遮阳效果。我们把数值大于 80%的路径称为达标路径。我们将这两项环境属性的阈值标准设定为 80%，当某条路径满足路灯保障率 $\geq 80\%$ 或树荫覆盖率 $\geq 80\%$ 时，我们认定为"达标路径"。

通过 MATLAB 平台进行仿真实验，我们在标准校园路网数据集上对比了遗传算法（GA）与经典 Dijkstra 算法的性能表现。实验结果对比如表 5-3 所示，图 5-2 为数据的柱状图表示。实验结果表明，遗传算法产生的路径中达标路径占比达到 87.6%，显著高于迪杰斯特拉算法的 63.2%，遗传算法在环境属性满足率上显著要优于迪杰斯特拉算法。这种优势源于遗传算法的群体智能特性，通过适应度函数中的加权求和策略能有效平衡多个目标函数的竞争关系，而迪杰斯特拉算法受限于单一路径代价评估机制，难以处理多目标耦合的优化问题。在多属性约束的校园路径规划场景中，遗传算法在环境适配性与多属性满足要求中有不可替代性和一定的优越性。



表 5-3 多属性保障率与达标路径占比实验数据对比

Table 5-3 Comparison of Experimental Data of Multi-attribute Guarantee Rate and the Proportion of Up-to-standard Path

场景	算法	平均保障率	达标路径占比 (>80%)
夜间	GA	92.3%	89.1%
	Dijkstra	41.7%	12.7%
白天	GA	88.6%	85.5%
	Dijkstra	35.2%	9.1%

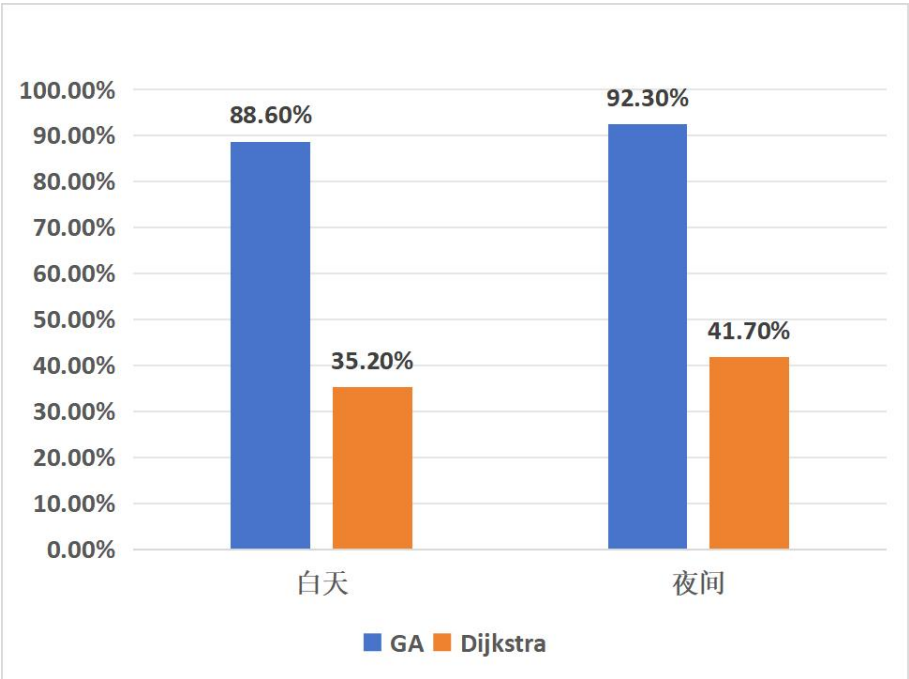


图 5-1 多属性保障率实验数据对比

Figure 5-1 Comparison of Experimental Data of Multi-attribute Guarantee Rate

5.2.2 基于遗传算法的方法结果分析

以夜晚模式的实验为例，我们从夜晚模式下的路径规划实验结果可以看出，路径图中频繁出现的中转节点（如节点 3、6、5）在路径网络图扮演重要枢纽角色，这现象表明这些中间节点具有良好的连通性和综合指标表现。例如，从节点 1 到节点 10 的路径为 1、3、6、10，该路径综合得分为 1.0514，结果兼顾了距离和路灯条件。这一现象主要源于它们的高连通性和综合指标优势。第一，高连通性指这些节点通常位于校园主干道交汇处，连接多个功能区（如教学区、宿舍区、食堂等），使得算法在优化路径时倾向于选择它们作为中转点。第二，综合指标表现指这些节点不仅具备较高的路灯覆盖率（保障夜间照明），同时路径距离适中，使得最终规划的路径在安全性和效率之间达到平衡。

从白天模式和夜晚模式的对比来看，白天模式更偏向于选择具有较高景观评分

与树荫覆盖的路径，这设计有利于保障最短距离的同时，提升路径舒适性与美观性。算法倾向于选择树荫覆盖率高（ $\geq 80\%$ ）的路径，例如经过绿化带、林荫道或景观湖的路线。这一设定与现实中人们在白天更偏好绿色遮荫、风景优美路径的行为倾向一致。夜晚模式则加强了对路灯数量的关注，这设计更倾向于选择光照充足和安全性更高的路线，算法更关注路灯覆盖率（ $\geq 80\%$ ）和路径可见度，避免选择偏僻或照明不足的路线。符合人们夜间行走时对照明条件要求。遗传算法在种群规模和迭代次数适当设置下，可稳定输出合理的解方案。合理设计路径的有效性判断与交叉变异机制，对于算法性能提升具有显著作用。

综上所述，本研究设计的多属性遗传算法具备良好的鲁棒性与适应性。算法能够根据不同时间段动态调整权重参数，实现路径选择策略的灵活变化性。相比传统 Dijkstra 算法（仅优化单一路径长度），遗传算法能够同时优化距离、照明、树荫、景观等多个目标，找到 Pareto 最优解集。基于遗传算法的路径规划模型在不同时间段下能够自适应做出相对应的调整，具有较好的优化效果和稳定性。

## 6 结论与展望

### 6.1 研究成果总结

第一，主要研究发现与贡献。本研究主要用图论建模与遗传算法，构建了一个适用于校园环境的多属性路径规划模型。算法结合了不同时间段引入动态权重机制，能有效地提升路径选择的适应性。实验结果表明，该方法在优化路径选择方面具有良好的实用性和适应性。

第二，研究过程中遇到的局限性。本研究主要针对校园环境进行路径规划，但由于数据来源较为单一，模型在应对复杂城市交通系统时可能存在局限性。此外，遗传算法在大规模交通网络中的计算复杂度较高，尤其在优化过程中可能面临效率瓶颈。

第三，算法的有效性与优越性。本研究通过引入多种环境因素及权重动态调整机制，让遗传算法在路径规划中体现出更高的环境感知能力。本研究设计的遗传算法与传统只考虑路程的路径规划方案相比，在用户体验与路径合理性方面具有明显优势。本研究设计的遗传算法通过其特有的选择、交叉与变异机制，有效提升了路径搜索的优化能力。相较于传统路径规划方法，遗传算法能够在更短的时间内收敛至较优解，有效地降低了因穷举式搜索所导致的计算开销与复杂度。本研究采用动态权重调整，根据白天和夜晚的不同需求自动调节路径选择，展示了其良好的适应性。算法能够在不同的时间段下选择最优路径，确保了结果的针对性和实际应用价值。

### 6.2 研究的应用前景

本研究创新性地结合了遗传算法与多属性路径规划，并在校园路径规划中融入了景观、树荫和路灯等因素。这设计可以使得用户在路径选择中不仅仅可以考虑最短距离，还能综合考虑环境因素的影响。

随着智能交通系统和智慧校园建设的发展，本研究提出的路径规划方法可以有效地应用于更广泛的如城市交通规划、智能导航等的场景。未来我们可以进一步改进算法的适应性和实时性，结合更多的如天气、交通状态等的动态因素，让路径规划更加精确。我们还可以结合深度学习等新技术，提升算法的智能化和高效性。在大规模数据处理和并行计算方面，我们还需要克服更大的挑战。企业未来可以通过实时数据的反馈和算法的调整，为用户提供更精准和个性化的路径规划服务。

## 参 考 文 献

- [1] 《中国公路学报》编辑部. 中国交通工程学术研究综述·2016[J]. 中国公路学报, 2016, 29(6): 1-161.
- [2] 崔京浩. 解析城市交通[M]. 北京: 中国水利水电出版社, 2001.
- [3] 张雨佳. 寒地大学校园步行建成环境对学生择路行为的影响研究[D]. 哈尔滨工业大学, 2023.
- [4] 湛文静, 李泳科. 基于改进遗传算法的路径规划问题相关研究综述[J]. 计算机与数字工程, 2023, 51(07): 1544-1550.
- [5] 曹大有, 马斌. 基于遗传算法的单源最短路径研究[J]. 汉江师范学院学报, 2021, 41(06): 1-5.
- [6] Gibson J, Schuler T, McGuire L, et al. Swarm and Multi-agent Time-based A\* Path Planning for LTA3 Systems[J]. Unmanned Systems, 2020, 08(03): 253-260.
- [7] Wang T, Huang P, Dong G. Modeling and Path Planning for Persistent Surveillance by Unmanned Ground Vehicle[J]. IEEE Transactions on Automation Ence and Engineering, 2020, 17(99): 1-11.
- [8] 潘长安. 基于改进 A 星算法的城市交通寻径的研究[D]. 华侨大学, 2015.
- [9] 赵艳丽. 实际路网最短路径算法优化与实现[D]. 华南理工大学, 2015.
- [10] 张得天, 王佳傲, 陈飞. 国内网络地图服务比较分析研究 (英文) [J]. 华东师范大学学报(自然科学版), 2017, 63(06): 85-95.
- [11] 毛彬. 美国大学校园道路交通景观设计探析[J]. 华中建筑, 2015, 33(09): 117-120.
- [12] 王栋. 基于改进 Dijkstra 算法的共享停车系统设计[D]. 南京信息工程大学, 2023.
- [13] 吴红波, 王英杰, 杨肖肖. 基于 Dijkstra 算法优化的城市交通路径分析[J]. 北京交通大学学报, 2019, 43(4): 116-121, 130.
- [14] 杜宗泽. 基于改进 A\* 算法的路径规划研究[D]. 中南大学, 2022.
- [15] Tao H, Wang Z, Li J. Three-dimensional path planning for unmanned aerial vehicles based on multi-objective genetic algorithm[C]//2014 33rd Chinese Control Conference(CCC). IEEE, 2014.
- [16] 王小平, 曹立明. 遗传算法 - 理论、应用与软件实现 [M]. 西安: 西安交通大学出版社, 2002: 7~15, 25~37.
- [17] 郭庆腾. 改进遗传算法求解车辆路径问题研究[D]. 青岛大学, 2023.
- [18] 屈兴科. 基于改进遗传算法的主题公园动态路径规划研究[D]. 哈尔滨商业大学, 2023.
- [19] 许占栋. 高德打造全域服务地图[J]. 中国测绘, 2025, 34(02): 52-53.
- [20] 董小帅, 毛政元. 基于改进遗传算法的动态路径规划研究[J]. 计算机工程与应用, 2018, 54(19): 49-55.
- [21] 杨从锐, 钱谦, 王锋, 等. 改进的自适应遗传算法在函数优化中的应用[J]. 计算机应用研究, 2018, 35(4): 1042-1045.

- [22] 孙娓娓,孙怀波,李俊林.基于 AHP-TOPSIS 模型的“数学建模”课程思政的成效分析[J].湖北理工学院学报,2025,41(02):73-77.
- [23] 孙睿,黄海超,霍欣婷,等.多目标优化-改进遗传算法路径规划模型[J].贵州大学学报(自然科学版),2023,40(01):95-100+124.
- [24] 程佳佳,张旭,张海霞,等.基于精英引导的遗传算法及路径规划应用[C]//中国自动化学会.2024 中国自动化大会论文集.江苏省邮政大数据技术与应用工程研究中心,国家邮政局邮政行业技术研发中心(物联网技术),南京邮电大学现代邮政学院;安徽邮谷快递智能科技有限公司;圆通速递股份有限公司,;2024:503-508.
- [25] WU D Q, LI J Y, CUI J Y, et al. Research on the time-dependent vehicle routing problem for fresh agricultural products based on customer value [J]. Agriculture, 2023, 13(3): 681.

## 附录一：遗传算法实验代码

```
1.    %% 参数设置
2.    pointnumber = 11; % 节点数量
3.    Popsize = 300; % 种群规模
4.    MaxGeneration = 150; % 最大迭代次数
5.    Pc = 0.8; % 交叉概率
6.    Pm = 0.15; % 变异概率
7.    eliteRatio = 0.1; % 精英保留比例
8.    % 初始化种群
9.    population = cell(Popsize, 1);
10.   for i = 1:Popsize
11.       population{i} = generateValidPath(A, start, goal);
12.   end
13.   best_score = -inf; best_path = [];
14.   for gen = 1:MaxGeneration
15.       % 评估种群
16.       scores = zeros(Popsize, 1);
17.       valid_flags = false(Popsize, 1);
18.       parfor i = 1:Popsize
19.           [valid_flags(i), scores(i)] = evaluatePath
20.           (population{i}, A, B, C, D, weights, min_dist, max_dist);
21.       end
22.       % 更新最优解
23.       [current_max, idx] = max(scores);
24.       if current_max > best_score
25.           best_score = current_max;
26.           best_path = population{idx};
27.       end
28.       % 精英选择
29.       eliteCount = round(Popsize * eliteRatio);
30.       [~, eliteIndices] = maxk(scores, eliteCount);
31.       new_population = population(eliteIndices);
```

```
32. % 轮盘赌选择
33. valid_indices = find(valid_flags);
34. if ~isempty(valid_indices)
35.     valid_scores = scores(valid_flags);
36.     valid_scores = valid_scores - min(valid_scores) + eps;
37.     prob = valid_scores / sum(valid_scores);
38.     cum_prob = cumsum(prob);
39.     for i = (eliteCount+1):Popsize
40.         r = rand();
41.         sel = find(cum_prob >= r, 1);
42.         if isempty(sel)
43.             sel = length(valid_indices);
44.         end
45.         new_population{i} = population{valid_indices(sel)};
46.     end
47. end
48. population = new_population;
49. % 交叉操作
50. for i = 1:2:Popsize-1
51.     if rand() < Pc
52.         [population{i}, population{i+1}] = ...
53.         crossoverPaths(population{i},
54.             population{i+1}, A, B, C, D, start, goal);
55.     end
56. end
57. for i = 1:Popsize % 变异操作
58.     if rand() < Pm
59.         population{i} = mutatePath(population{i}, A, start, goal);
60.     end
61. end
62. % 精英保留
63. population{1} = best_path;
64. end
```

附录二：遗传算法实验结果

=== 最优路径结果 ===		
时间：夜晚		
权重：距离(0.577)，景观(0.115)，路灯(0.192)，树荫(0.115)		
从 1 到 2: 1 → 2		综合得分: 1.7093
从 1 到 3: 1 → 3		综合得分: 1.7048
从 1 到 4: 1 → 4		综合得分: 1.5358
从 1 到 5: 1 → 2 → 5		综合得分: 1.4591
从 1 到 6: 1 → 3 → 6		综合得分: 1.2572
从 1 到 7: 1 → 3 → 7		综合得分: 1.3363
从 1 到 8: 1 → 3 → 6 → 8		综合得分: 1.1946
从 1 到 9: 1 → 3 → 6 → 9		综合得分: 1.0064
从 1 到 10: 1 → 3 → 6 → 10		综合得分: 1.0514
从 1 到 11: 1 → 3 → 6 → 9 → 11		综合得分: 0.7147
从 2 到 1: 2 → 1		综合得分: 1.7093
从 2 到 3: 2 → 3		综合得分: 1.6710
从 2 到 4: 2 → 3 → 4		综合得分: 1.5065
从 2 到 5: 2 → 5		综合得分: 1.5763
从 2 到 6: 2 → 5 → 6		综合得分: 1.4050
从 2 到 7: 2 → 3 → 7		综合得分: 1.3599
从 2 到 8: 2 → 5 → 8		综合得分: 1.0250
从 2 到 9: 2 → 5 → 6 → 9		综合得分: 1.1413
从 2 到 10: 2 → 5 → 6 → 10		综合得分: 1.1864
从 2 到 11: 2 → 5 → 8 → 11		综合得分: 0.5479
从 3 到 1: 3 → 1		综合得分: 1.7048
从 3 到 2: 3 → 2		综合得分: 1.6710
从 3 到 4: 3 → 4		综合得分: 1.7195
从 3 到 5: 3 → 5		综合得分: 1.4937
从 3 到 6: 3 → 6		综合得分: 1.3404
从 3 到 7: 3 → 7		综合得分: 1.4580
从 3 到 8: 3 → 6 → 8		综合得分: 1.3609



从 3 到 9: 3 → 6 → 9	综合得分: 1.1727
从 3 到 10: 3 → 6 → 10	综合得分: 1.2178
从 3 到 11: 3 → 6 → 9 → 11	综合得分: 0.8779
从 4 到 1: 4 → 1	综合得分: 1.5358
从 4 到 2: 4 → 3 → 2	综合得分: 1.5065
从 4 到 3: 4 → 3	综合得分: 1.7195
从 4 到 5: 4 → 3 → 5	综合得分: 1.3677
从 4 到 6: 4 → 3 → 6	综合得分: 1.2144
从 4 到 7: 4 → 3 → 7	综合得分: 1.2935
从 4 到 8: 4 → 3 → 6 → 8	综合得分: 1.1326
从 4 到 9: 4 → 3 → 6 → 9	综合得分: 0.9444
从 4 到 10: 4 → 3 → 6 → 10	综合得分: 0.9895
从 4 到 11: 4 → 3 → 6 → 9 → 11	综合得分: 0.6431
从 5 到 1: 5 → 2 → 1	综合得分: 1.4591
从 5 到 2: 5 → 2	综合得分: 1.5763
从 5 到 3: 5 → 3	综合得分: 1.4937
从 5 到 4: 5 → 3 → 4	综合得分: 1.3677
从 5 到 6: 5 → 6	综合得分: 1.6552
从 5 到 7: 5 → 6 → 7	综合得分: 1.6214
从 5 到 8: 5 → 8	综合得分: 1.1217
从 5 到 9: 5 → 6 → 9	综合得分: 1.3915
从 5 到 10: 5 → 6 → 10	综合得分: 1.4366
从 5 到 11: 5 → 8 → 11	综合得分: 0.6957
从 6 到 1: 6 → 3 → 1	综合得分: 1.2572
从 6 到 2: 6 → 5 → 2	综合得分: 1.4050
从 6 到 3: 6 → 3	综合得分: 1.3404
从 6 到 4: 6 → 3 → 4	综合得分: 1.2144
从 6 到 5: 6 → 5	综合得分: 1.6552
从 6 到 7: 6 → 7	综合得分: 1.9077
从 6 到 8: 6 → 8	综合得分: 1.8660
从 6 到 9: 6 → 9	综合得分: 1.6778
从 6 到 10: 6 → 10	综合得分: 1.7229
从 6 到 11: 6 → 9 → 11	综合得分: 1.2935
从 7 到 1: 7 → 3 → 1	综合得分: 1.3363

从 7 到 2: 7 → 3 → 2	综合得分: 1.3599
从 7 到 3: 7 → 3	综合得分: 1.4580
从 7 到 4: 7 → 3 → 4	综合得分: 1.2935
从 7 到 5: 7 → 6 → 5	综合得分: 1.6214
从 7 到 6: 7 → 6	综合得分: 1.9077
从 7 到 8: 7 → 6 → 8	综合得分: 1.7747
从 7 到 9: 7 → 6 → 9	综合得分: 1.5865
从 7 到 10: 7 → 10	综合得分: 1.4220
从 7 到 11: 7 → 10 → 11	综合得分: 0.9667
从 8 到 1: 8 → 5 → 2 → 1	综合得分: 0.9207
从 8 到 2: 8 → 5 → 2	综合得分: 1.0250
从 8 到 3: 8 → 6 → 3	综合得分: 1.3609
从 8 到 4: 8 → 6 → 3 → 4	综合得分: 1.1326
从 8 到 5: 8 → 5	综合得分: 1.1217
从 8 到 6: 8 → 6	综合得分: 1.8660
从 8 到 7: 8 → 6 → 7	综合得分: 1.7747
从 8 到 9: 8 → 9	综合得分: 1.5414
从 8 到 10: 8 → 6 → 10	综合得分: 1.5899
从 8 到 11: 8 → 11	综合得分: 1.2085
从 9 到 1: 9 → 6 → 3 → 1	综合得分: 1.0064
从 9 到 2: 9 → 6 → 5 → 2	综合得分: 1.1413
从 9 到 3: 9 → 6 → 3	综合得分: 1.1727
从 9 到 4: 9 → 6 → 3 → 4	综合得分: 0.9444
从 9 到 5: 9 → 6 → 5	综合得分: 1.3915
从 9 到 6: 9 → 6	综合得分: 1.6778
从 9 到 7: 9 → 6 → 7	综合得分: 1.5865
从 9 到 8: 9 → 8	综合得分: 1.5414
从 9 到 10: 9 → 10	综合得分: 1.6034
从 9 到 11: 9 → 11	综合得分: 1.4997
从 10 到 1: 10 → 6 → 3 → 1	综合得分: 1.0514
从 10 到 2: 10 → 6 → 5 → 2	综合得分: 1.1864
从 10 到 3: 10 → 6 → 3	综合得分: 1.2178
从 10 到 4: 10 → 6 → 3 → 4	综合得分: 0.9895
从 10 到 5: 10 → 6 → 5	综合得分: 1.4366

从 10 到 6: 10 → 6	综合得分: 1.7229
从 10 到 7: 10 → 7	综合得分: 1.4220
从 10 到 8: 10 → 6 → 8	综合得分: 1.5899
从 10 到 9: 10 → 9	综合得分: 1.6034
从 10 到 11: 10 → 11	综合得分: 1.4862
从 11 到 1: 11 → 8 → 5 → 2 → 1	综合得分: 0.4244
从 11 到 2: 11 → 8 → 5 → 2	综合得分: 0.5479
从 11 到 3: 11 → 9 → 6 → 3	综合得分: 0.8779
从 11 到 4: 11 → 9 → 6 → 3 → 4	综合得分: 0.6431
从 11 到 5: 11 → 8 → 5	综合得分: 0.6957
从 11 到 6: 11 → 9 → 6	综合得分: 1.2935
从 11 到 7: 11 → 9 → 6 → 7	综合得分: 1.2405
从 11 到 8: 11 → 8	综合得分: 1.2085
从 11 到 9: 11 → 9	综合得分: 1.4997
从 11 到 10: 11 → 10	综合得分: 1.4862

### 附录三：Dijkstra 实验代码

```
1.    %% Dijkstra 多目标优化函数
2.    function [optimal_path, optimal_score] = dijkstra_multiobjective(A,
    B, C, D, weights, min_dist, max_dist, start, goal)
3.    n = size(A, 1);
4.    visited = false(1, n);
5.    distance = inf(1, n);
6.    previous = zeros(1, n);
7.    score = -inf(1, n);
8.    distance(start) = 0;
9.    score(start) = 0;
10.   for i = 1:n
11.   % 找到未访问节点中距离最小的节点
12.   current = -1;
13.   min_d = inf;
14.   for v = 1:n
15.   if ~visited(v) && distance(v) < min_d
16.   min_d = distance(v);
17.   current = v;
18.   end
19.   end
20.   if current == -1 || current == goal
21.   break;
22.   end
23.   visited(current) = true;
24.   % 更新邻居节点
25.   for neighbor = 1:n
26.   if A(current, neighbor) < inf && ~visited(neighbor)
27.   % 计算新路径的得分
28.   new_dist = distance(current) + A(current, neighbor);
29.   % 计算多目标得分
30.   norm_dist = 1 - (new_dist - min_dist)/(max_dist - min_dist);
```

```
31. norm_scenery = (B(current, neighbor) + B(neighbor, current))/2;
32. norm_light = (C(current, neighbor) + C(neighbor, current))/2;
33. norm_shade = (D(current, neighbor) + D(neighbor, current))/2;
34. new_score = weights(1)*norm_dist + weights(2)*norm_scenery + ...
35. weights(3)*norm_light + weights(4)*norm_shade;
36. % 如果新路径更好, 则更新
37. if new_score > score(neighbor)
38. distance(neighbor) = new_dist;
39. score(neighbor) = new_score;
40. previous(neighbor) = current;
41. end
42. end
43. end
44. end
45. % 重构最优路径
46. if isinf(distance(goal))
47. optimal_path = [start];
48. optimal_score = -inf;
49. else
50. path = [];
51. u = goal;
52. while u ~= 0
53. path = [u path];
54. u = previous(u);
55. end
56. optimal_path = path;
57. optimal_score = score(goal);
58. end
59. end
```

## 附录四：Dijkstra 实验结果

=== Dijkstra 算法最优路径结果 ===

时间：夜晚

权重：距离(0.577)，景观(0.115)，路灯(0.192)，树荫(0.115)

从 1 到 2: 1 → 2	综合得分: 1.7093
从 1 到 3: 1 → 3	综合得分: 1.7048
从 1 到 4: 1 → 3 → 4	综合得分: 1.5403
从 1 到 5: 1 → 2 → 5	综合得分: 1.4016
从 1 到 6: 1 → 3 → 7 → 6	综合得分: 1.4175
从 1 到 7: 1 → 3 → 7	综合得分: 1.2788
从 1 到 8: 1 → 3 → 7 → 6 → 8	综合得分: 1.2845
从 1 到 9: 1 → 3 → 7 → 6 → 9	综合得分: 1.0963
从 1 到 10: 1 → 3 → 7 → 6 → 10	综合得分: 1.1414
从 1 到 11: 1 → 3 → 7 → 6 → 10 → 11	综合得分: 0.6286
从 2 到 1: 2 → 1	综合得分: 1.7093
从 2 到 3: 2 → 3	综合得分: 1.6710
从 2 到 4: 2 → 3 → 4	综合得分: 1.6215
从 2 到 5: 2 → 5	综合得分: 1.5763
从 2 到 6: 2 → 3 → 7 → 6	综合得分: 1.4986
从 2 到 7: 2 → 3 → 7	综合得分: 1.3599
从 2 到 8: 2 → 3 → 7 → 6 → 8	综合得分: 1.3657
从 2 到 9: 2 → 3 → 7 → 6 → 9	综合得分: 1.1775
从 2 到 10: 2 → 3 → 7 → 6 → 10	综合得分: 1.2225
从 2 到 11: 2 → 3 → 7 → 6 → 10 → 11	综合得分: 0.7097
从 3 到 1: 3 → 1	综合得分: 1.7048
从 3 到 2: 3 → 2	综合得分: 1.6710
从 3 到 4: 3 → 4	综合得分: 1.7195
从 3 到 5: 3 → 5	综合得分: 1.4937
从 3 到 6: 3 → 7 → 6	综合得分: 1.5967
从 3 到 7: 3 → 7	综合得分: 1.4580
从 3 到 8: 3 → 7 → 6 → 8	综合得分: 1.4637

从 3 到 9: 3 → 7 → 6 → 9	综合得分: 1.2755
从 3 到 10: 3 → 7 → 6 → 10	综合得分: 1.3206
从 3 到 11: 3 → 7 → 6 → 10 → 11	综合得分: 0.8078
从 4 到 1: 4 → 1	综合得分: 1.5358
从 4 到 2: 4 → 3 → 2	综合得分: 1.3915
从 4 到 3: 4 → 3	综合得分: 1.7195
从 4 到 5: 4 → 3 → 5	综合得分: 1.2142
从 4 到 6: 4 → 3 → 7 → 6	综合得分: 1.3172
从 4 到 7: 4 → 3 → 7	综合得分: 1.1785
从 4 到 8: 4 → 3 → 7 → 6 → 8	综合得分: 1.1842
从 4 到 9: 4 → 3 → 7 → 6 → 9	综合得分: 0.9960
从 4 到 10: 4 → 3 → 7 → 6 → 10	综合得分: 1.0411
从 4 到 11: 4 → 3 → 7 → 6 → 10 → 11	综合得分: 0.5283
从 5 到 1: 5 → 2 → 1	综合得分: 1.5166
从 5 到 2: 5 → 2	综合得分: 1.5763
从 5 到 3: 5 → 3	综合得分: 1.4937
从 5 到 4: 5 → 3 → 4	综合得分: 1.5212
从 5 到 6: 5 → 6	综合得分: 1.6552
从 5 到 7: 5 → 6 → 7	综合得分: 1.6789
从 5 到 8: 5 → 6 → 8	综合得分: 1.6372
从 5 到 9: 5 → 6 → 9	综合得分: 1.4490
从 5 到 10: 5 → 6 → 10	综合得分: 1.4941
从 5 到 11: 5 → 6 → 10 → 11	综合得分: 0.9813
从 6 到 1: 6 → 7 → 3 → 1	综合得分: 1.3025
从 6 到 2: 6 → 5 → 2	综合得分: 1.3475
从 6 到 3: 6 → 7 → 3	综合得分: 1.3667
从 6 到 4: 6 → 7 → 3 → 4	综合得分: 1.3172
从 6 到 5: 6 → 5	综合得分: 1.6552
从 6 到 7: 6 → 7	综合得分: 1.9077
从 6 到 8: 6 → 8	综合得分: 1.8660
从 6 到 9: 6 → 9	综合得分: 1.6778
从 6 到 10: 6 → 10	综合得分: 1.7229
从 6 到 11: 6 → 10 → 11	综合得分: 1.2101
从 7 到 1: 7 → 3 → 1	综合得分: 1.3938

从 7 到 2: 7 → 3 → 2	综合得分: 1.3599
从 7 到 3: 7 → 3	综合得分: 1.4580
从 7 到 4: 7 → 3 → 4	综合得分: 1.4085
从 7 到 5: 7 → 6 → 5	综合得分: 1.5639
从 7 到 6: 7 → 6	综合得分: 1.9077
从 7 到 8: 7 → 6 → 8	综合得分: 1.7747
从 7 到 9: 7 → 6 → 9	综合得分: 1.5865
从 7 到 10: 7 → 6 → 10	综合得分: 1.6316
从 7 到 11: 7 → 6 → 10 → 11	综合得分: 1.1188
从 8 到 1: 8 → 6 → 7 → 3 → 1	综合得分: 1.1695
从 8 到 2: 8 → 6 → 5 → 2	综合得分: 1.2145
从 8 到 3: 8 → 6 → 7 → 3	综合得分: 1.2337
从 8 到 4: 8 → 6 → 7 → 3 → 4	综合得分: 1.1842
从 8 到 5: 8 → 6 → 5	综合得分: 1.5222
从 8 到 6: 8 → 6	综合得分: 1.8660
从 8 到 7: 8 → 6 → 7	综合得分: 1.7747
从 8 到 9: 8 → 6 → 9	综合得分: 1.5448
从 8 到 10: 8 → 6 → 10	综合得分: 1.5899
从 8 到 11: 8 → 11	综合得分: 1.2085
从 9 到 1: 9 → 6 → 7 → 3 → 1	综合得分: 0.9813
从 9 到 2: 9 → 6 → 5 → 2	综合得分: 1.0263
从 9 到 3: 9 → 6 → 7 → 3	综合得分: 1.0455
从 9 到 4: 9 → 6 → 7 → 3 → 4	综合得分: 0.9960
从 9 到 5: 9 → 6 → 5	综合得分: 1.3340
从 9 到 6: 9 → 6	综合得分: 1.6778
从 9 到 7: 9 → 6 → 7	综合得分: 1.5865
从 9 到 8: 9 → 8	综合得分: 1.5414
从 9 到 10: 9 → 10	综合得分: 1.6034
从 9 到 11: 9 → 11	综合得分: 1.4997
从 10 到 1: 10 → 6 → 7 → 3 → 1	综合得分: 1.0264
从 10 到 2: 10 → 6 → 5 → 2	综合得分: 1.0714
从 10 到 3: 10 → 6 → 7 → 3	综合得分: 1.0906
从 10 到 4: 10 → 6 → 7 → 3 → 4	综合得分: 1.0411
从 10 到 5: 10 → 6 → 5	综合得分: 1.3791



从 10 到 6: 10 → 6	综合得分: 1.7229
从 10 到 7: 10 → 6 → 7	综合得分: 1.6316
从 10 到 8: 10 → 6 → 8	综合得分: 1.5899
从 10 到 9: 10 → 9	综合得分: 1.6034
从 10 到 11: 10 → 11	综合得分: 1.4862
从 11 到 1: 11 → 9 → 6 → 7 → 3 → 1	综合得分: 0.7120
从 11 到 2: 11 → 9 → 6 → 5 → 2	综合得分: 0.7570
从 11 到 3: 11 → 9 → 6 → 7 → 3	综合得分: 0.7762
从 11 到 4: 11 → 9 → 6 → 7 → 3 → 4	综合得分: 0.7267
从 11 到 5: 11 → 9 → 6 → 5	综合得分: 1.0647
从 11 到 6: 11 → 9 → 6	综合得分: 1.4085
从 11 到 7: 11 → 9 → 6 → 7	综合得分: 1.3172
从 11 到 8: 11 → 9 → 8	综合得分: 1.2720
从 11 到 9: 11 → 9	综合得分: 1.4997
从 11 到 10: 11 → 10	综合得分: 1.4862

## 致 谢

时间过得真快，我本科学习时间是在汗水和拼搏中度过的，在广东财经大学学习的这些年里，我学到了许多专业知识，开阔了视野，同时，也锻炼了意志，对于未来，也多了许多的憧憬。在论文的完成过程中，得到了导师的大力帮助，从论文的选题、开题、写作、修改到最终完成，自始至终得到了老师们严格、详细和耐心的指导，他们为我论文的完成倾注了大量的心血。在论文撰写过程中，他们严谨的治学态度、认真的工作精神和求实勤勉的工作作风令我由衷敬佩并受益终身，在此，我要向辛勤培养我的导师们致以深深的感谢和崇高的敬意！

感谢所有帮助过我的老师、同学和朋友，感谢广东财经大学，感谢在我工作和生活中每一位曾经帮助过我的人，没有他们的大力支持就没有我的今天，我也知道我永远都无法回报他们的爱，所以我希望顺利地完学业，给他们一点点慰藉。

同时，我要感谢我挚爱的家人，是他们的支持与理解给了我继续深造的勇气，是他们的关怀与呵护给了我学习的动力，使我能够专心致志的完成本科学业。

最后，祝愿所有关心过我和我关心的人心想事成，万事如意！