

基于遗传算法的配 送路线优化系统



——项目详细方案

汇报小组：HAPDEAN

目录

1.	引言.....	3
1.1	编写目的.....	3
1.2	背景.....	3
1.3	定义.....	3
1.4	参考资料.....	4
2.	目标与解决思路.....	5
2.1	赛题价值.....	5
2.1.1	项目背景.....	5
2.1.2	项目概述.....	7
2.1.3	项目意义.....	7
2.2	解决思路.....	8
2.2.1	总体思路.....	8
2.2.2	系统总体功能.....	8
2.2.3	系统流程图.....	9
2.2.4	数据流图.....	9
3.	问题与解决方案.....	10
3.1	系统用例分析.....	10
3.1.1	管理员用例分析.....	10
3.1.2	普通用户用例分析.....	12
3.2	非功能性需求.....	14
3.2.1	应用兼容性.....	14
3.2.2	性能需求.....	14
3.2.3	可靠性需求.....	14
3.2.4	可用性需求.....	14
3.2.5	出错处理需求.....	14
3.2.6	约束.....	15
3.2.7	逆向需求.....	15
3.2.8	将来可能提出的要求.....	15
3.2.9	可扩充性.....	15
3.2.10	可维护性.....	15
3.2.11	可移植性.....	15
3.2.12	美观性.....	15
3.2.13	易用性.....	16
3.2.14	正确性.....	16
3.2.15	健壮性.....	16
3.2.16	清晰性.....	16
3.2.17	安全性.....	16
3.3	应用对象.....	16
3.4	应用环境.....	17
4.	技术路线及设计方案.....	18
4.1	技术路线.....	18
4.2	模型设计.....	21

4.3	算法设计	23
4.3.1	染色体编码	23
4.3.2	初始种群的确定	24
4.3.3	适应度函数的设定	24
4.3.4	遗传算法的设计	24
4.3.5	交叉算子和变异算子的改进	25
5.	项目管理与业务模式	26
5.1	项目计划	26
5.2	项目估算	27
5.3	角色分配与职责	27
5.4	开发管理	28
5.5	项目监控	29
5.5.1	项目评审	29
5.5.2	项目过程与质量管理	30
5.5.3	项目沟通与风险管理	31
5.6	成本分析	31
5.7	市场与竞争分析	34
5.8	风险控制	36
6.	技术可行分析	39
6.1	概述	39
6.2	技术不足风险	39
6.2.1	面对节点数量大的稠密图，求解时间较长	39
6.2.3	对于非法输入缺乏限制，导致程序出错	40
6.2.4	在极端情况下，有极小几率出现非最优解	40
6.3	技术使用风险	40
6.3.1	路线图存储风险	40
6.4	易用性及使用门槛	41
6.5	产品环境依赖性	41
6.6	技术要求	41
7.	其他可行性分析	42
7.1	经济可行性	42
7.2	社会可行性	42
7.3	实用可行性	43
7.4	法律可行性	43
7.5	人员可行性	43

1. 引言

1.1 编写目的

本项目详细方案，是在概要设计说明书的基础上进一步明确系统结构，详细的说明项目的开发过程与人员，并对该项目进行详细的说明与分析。

1.2 背景

随着信息技术的发展，现代物流作为“第三个利润源泉”是一种先进的组织方式和管理技术，已被各国广泛采用，并形成产业化，在国民经济中发挥越来越重要的作用，这一管理技术，正受到日益广泛的重视，并面临巨大的发展机会，在现代物流中，配送是一个重要的与消费者直接相连的环节，一方面可以体现企业的核心竞争力，另一方面，通过线路优化，可以提高企业的运作效率，降低配送成本，实现物流科学化。目前市场上仓储类管理系统已经比较成熟完善，但对于不同需求的路线规划还存在着一定的短板，基于当前站好“最后一岗”的要求，引用路线优化算法，开发出具有实用性，先进性，高可靠性的路线规划系统。

1.3 定义

Visual Studio 2017: 该软件是美国微软公司开发的 IDE，集成了开发 web 手机应用，嵌入式，游戏等环境是一个强大的集成开发环境，使用 vs 可以更容易管理工程把主要精力放到代码实现上，使用 vs 可以更加便利的开发 C/C++ 语言程序。

QT: 一个 1991 年由 Qt Company 开发的跨平台 C++ 图形用户界面应用程序开发框架。它既可以开发 GUI 程序，也可用于开发非 GUI 程序，比如控制台工具和服务器。

弗洛伊德算法: 是一种利用动态规划的思想寻找给定的加权图中多源点之间最短路径的算法，与 Dijkstra 算法类似。该算法名称以创始人之一、1978 年图灵奖获得者、斯坦福大学计算机科学系教授罗伯特·弗洛伊德命名。

遗传算法: 遗传算法 (Genetic Algorithm, GA) 最早是由美国的 John holland 于 20 世纪 70 年代提出, 该算法是根据大自然中生物体进化规律而设计提出的。是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型, 是一种通过模拟自然进化过程搜索最优解的方法。该算法通过数学的方式, 利用计算机仿真运算, 将问题的求解过程转换成类似生物进化中的染色体基因的交叉、变异等过程。在求解较为复杂的组合优化问题时, 相对一些常规的优化算法, 通常能够较快地获得较好的优化结果。

1.4 参考资料

[1] 《分布式并行遗传算法求解多车型车辆路径问题》王超，袁杰红（国防科技大学 空天科学学院，湖南 长沙 410073）

[2] 《基于改进自适应遗传算法的物流 配送路径优化研究》吴聪，陈侃松，姚静（湖北大学 计算机与信息工程学院物联网工程研究所，武汉 430062）

[3] 《基于节约里程法的配送路线优化》
研究 一以苏宁电器为例 周磊（南京师范大学，江苏 210023）

[4] 《煤矿物资多车型配送的改进遗传算法求解》郭海湘 1， 杨娟 2， 马争艳 3， 李兰兰 1（1. 中国地质大学经济管理学院，湖北武汉 430074；2. 西安交通大学管理学院，陕西西安 710049；3. 湖北宏观经济研究所，湖北武汉 430074）

[5] 《新型访问域部分重叠的多旅行商问题的 G A 求解》
孙启瑞 李俊 丁健 戴先中（东南大学 复杂工程 系统 测量与控制教育部重点实验室 江苏 南京 210096）

[6] 吴聪,陈侃松,姚静. 基于改进自适应遗传算法的物流配送路径优化研究[J]. 计算机测量与控制, 2018, 26 (2)

2. 目标与解决思路

2.1 赛题价值

2.1.1 项目背景

路径规划在很多领域都具有广泛的应用。在高新科技领域的应用有：机器人的自主无碰行动；无人机的避障突防飞行；巡航导弹躲避雷达搜索、防反弹袭击、完成突防爆破任务等。在日常生活领域的应用有：GPS 导航；基于 GIS 系统的道路规划；城市道路网规划导航等。在决策管理领域的应用有：物流管理中的车辆问题 (VRP) 及类似的资源管理资源配置问题。通信技术领域的路由问题等。凡是可拓扑为点线网络的规划问题基本上都可以采用路径规划的方法解决。随着机器人 SLAM 的发展以及物流系统的进步，路径规划逐渐成为老生常谈的技术问题。

根据对环境信息的把握程度可把路径规划划分为基于先验完全信息的全局路径规划和基于传感器信息的局部路径规划。其中，从获取障碍物信息是静态或是动态的角度看，全局路径规划属于静态规划 (又称离线规划)，局部路径规划属于动态规划 (又称在线规划)。全局路径规划需要掌握所有的环境信息，根据环境地图的所有信息进行路径规划；局部路径规划只需要由传感器实时采集环境信息，了解环境地图信息，然后确定出所在地图的位置及其局部的障碍物分布情况，从而可以选出从当前结点到某一子目标结点的最优路径。

根据所研究环境的信息特点，路径规划还可分为离散域范围内的路径规划问题和连续域范围内的路径规划问题。离散域范围内的路径规划问题属于一维静态优化问题，相当于环境信息简化后的路线优化问题；而连续域范围内的路径规划问题则是连续性多维动态环境下的问题。

据了解，传统的配送路径优化系统存在以下问题，也是我们需要攻克的问题：

(1) 计算效率低

算法的复杂程度决定了程序的运行速度，而路径规划往往需要用到很多数据结构，数据结构使用的不合理会导致程序运行过慢。

(2) 结果准确度低

很多路径规划系统舍弃了准确度而去追求运行速度，这在很大程度上对用户的需求大打折扣。想要兼顾速度和准确度必须在代码设计和参数调整上下足功夫。

(3) 鲁棒性差

算法的鲁棒性决定了它的可扩展性，当我们只使用单一的算法很难得到想要的结果。一个好的系统应该容纳多个算法，多个算法紧密配合，互相协调，最终

才能得到接近理想的结果。

（4）数学模型粗糙

一个算法的好坏得益于数学建模的功劳，数学模型是算法的核心，一个路径规划算法如果没有全面的数学模型就很容易陷入局部最优解。解决办法就是通过人工操作不停地构想和实验，多次改进数学模型，让算法准确度上升。

同时，考虑到本软件的应用环境应该包含农村和城市，又存在以下几点问题需要我们思考研究：

（1）路口的处理问题

在地图中存在路口节点的情况，应该对所有节点进行分类，并在算法内部对不同的节点实行不同的操作，比如路口节点只用来经过，而配送中心和配送点用来卸货装货或者经过，这样的设计使得软件应用环境更广泛更切合实际，同时算法的多样性也得到体现。

（2）路线最优问题

关于如何在成本最小，载重率尽可能大，车辆用数尽可能少，车辆运行的里程中进行权衡以及如何根据遗传算法的结果再次加工得到最短路径是我们需要关心的问题。首先遗传算法得出的结果是理论上的最短路径，但存在一定的偶然性，这就需要我们后期加入其他算法，比如迪杰斯特拉算法来对初步结果进行再次求解，这样既保证了一开始需求的最“短”也保证了距离上的最短。

关于如何在距离和成本上进行权衡，分析得出距离和成本是成正比的，我们一般只需要求出距离最短便可大致得出成本最小的方案，同时本软件给出了权重的设置选项，用户也可以根据用户操作手册设置这些参数，实现不同的需求。

（3）运行精度和运行时间的问题

算法的运行时间和精度一样重要，但在不同的应用场景下需要不同的软件性能，比如小型的项目可能更需要速度，而大型的项目则需要更加精确的方案，本软件决定倾向于软件精度，其次通过多次测试得出最短运行时间，保证精度和时间的最大化。

2.1.2 项目概述

本项目属于全局路径规划问题，即在已知所有地图信息的基础上进行路径规划，根据当前传统的路径规划算法存在的问题，我们提出了以下应用策略和开发思路：

（1）遗传算法的优化

为了解决“最优路径”的问题，本软件决定采用 GA 算法，然后再遗传算子的选择和变异算子的选择上选则了更为良好的自适应算法，防止遗传算法进入局部最优解，导致算法得出的结论不够准确。

（2）数据透明化，可操作化

将算法中用到的一些变量透明化，给予用户修改和设置的权限，用户可根据操作手册设置这些变量，程序会展示出不同的结果，提高了程序的泛用性。

（3）UI 界面

开发一个供用户简单操作的 UI 界面，主要包含数据创建、数据保存、数据读取、数据修改、使用说明、运行程序等功能，极大地方便了用户理解和使用该程序。

2.1.3 项目意义

（1）对企业的意义

企业使用该软件后，所有物流数据可以不通过联网得到处理，保证了公司的安全，同时企业通过选择路线的权重，可以得出更因地制宜的配送路线和配送方式，方便了物流企业配送货物，节约了很多不必要的成本，提高了企业的核心竞争力。同时该项目可以选择许多配送中心的信息，更为合理的贴合了企业的实际需要。配送环节是一个重要的与消费者相关联的环节，配送运输的快慢，是影响消费者对物流企业的评价的重要因数，所以本项目尽可能的在工作时间内完成地图中所有配送点的运输，从一定意义上也能提高物流配送的速度和效率，让物流企业能够给消费者留下一个好的印象，帮助提升企业的形象与口碑。

（2）对社会的意义

现代物流的发展趋势越来越趋于智能化，所以我们的项目可以推进物流配送路线的设计及优化，努力实现“站好最后一岗”。一方面配送路线的优化，关乎配送车辆的成本，节约成本意味着有更多的资金可以用于其他方面，更好的帮助企业的发展。另一方面减少车辆排放和损耗，在一定意义上也对环保有促进作用。还有一方面就是车辆配送次数的减少，一定程度下可以缓解交通的拥挤，减轻了交通的压力和降低发生车祸的概率。

2.2 解决思路

2.2.1 总体思路

关于路线规划系统的设计思路如下：

(1) 信息输入自由化：对于配送的各种信息，诸如车辆数量，车载，时间等由用户自由控制。

(2) 批量导入和单体创建：用户可以直接导入已有数据，也可以自己创建数据文件。

(3) 数据修改简单化：用户可以直接在修改界面上修改数据而不需要在数据库中修改。

(4) 权重分配自由组合：对于客户，可根据自己的偏好选择自己的目标函数。

2.2.2 系统总体功能

总体功能分为 7 大模块：创建数据模块，加载数据模块，修改数据模块，信息输入模块，权重分配模块，路线规划模块以及路线展示模块。

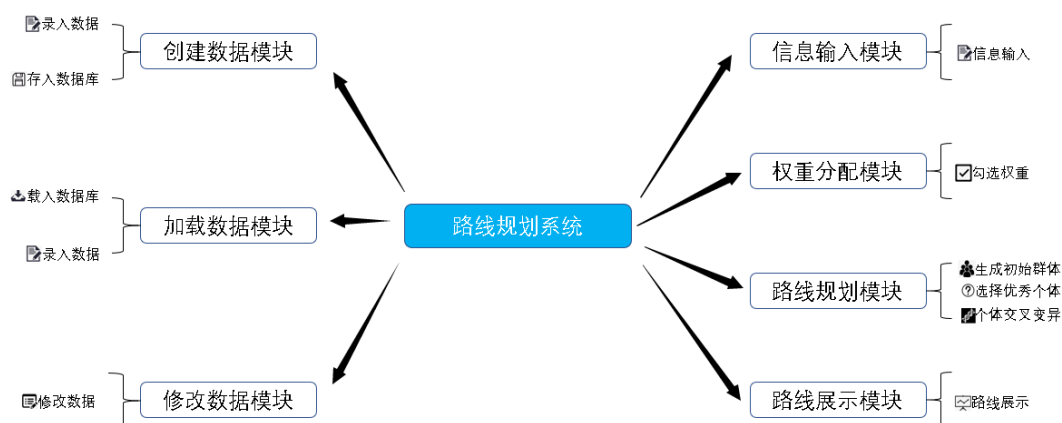


图 2.2.2 系统总体功能图

2.2.3 系统流程图

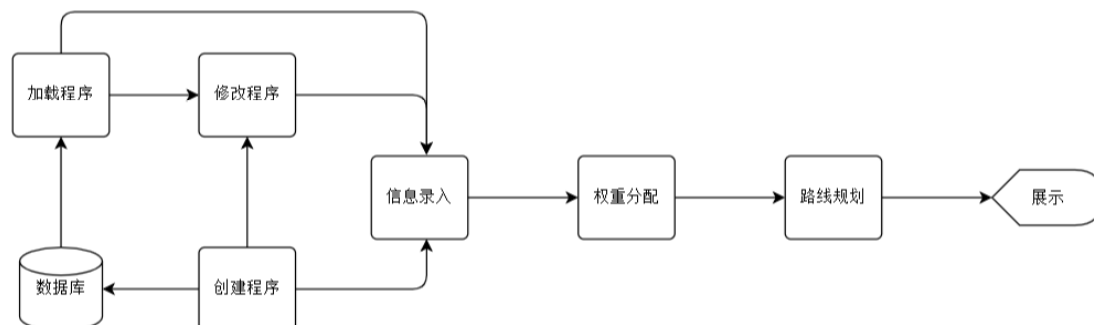


图 2.2.3 系统流程图

2.2.4 数据流图

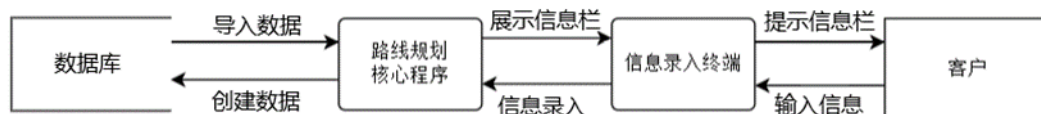


图 2.2.4 数据流图

3. 问题与解决方案

3.1 系统用例分析

所谓用例分析，就是从一个运用我们系统的用户的视角分析整个软件系统能够提供的功能，以及这些功能到底是提供给哪些角色使用。我们在用例分析时使用 ProcessOn 绘制用例图，通过用例建模来对系统进行功能和角色方面的梳理和分析。

通过对应用的需求调研后，分析得到配送路径优化系统可能面对的用户有：管理员、普通用户共两种，普通用户只能通过 UI 界面使用本系统，而管理员则可以使用 VS 等 IDE 深入算法部分对算法模型中的参数进行修改。

3.1.1 管理员用例分析

管理员用例图如图 3.1.1 所示：

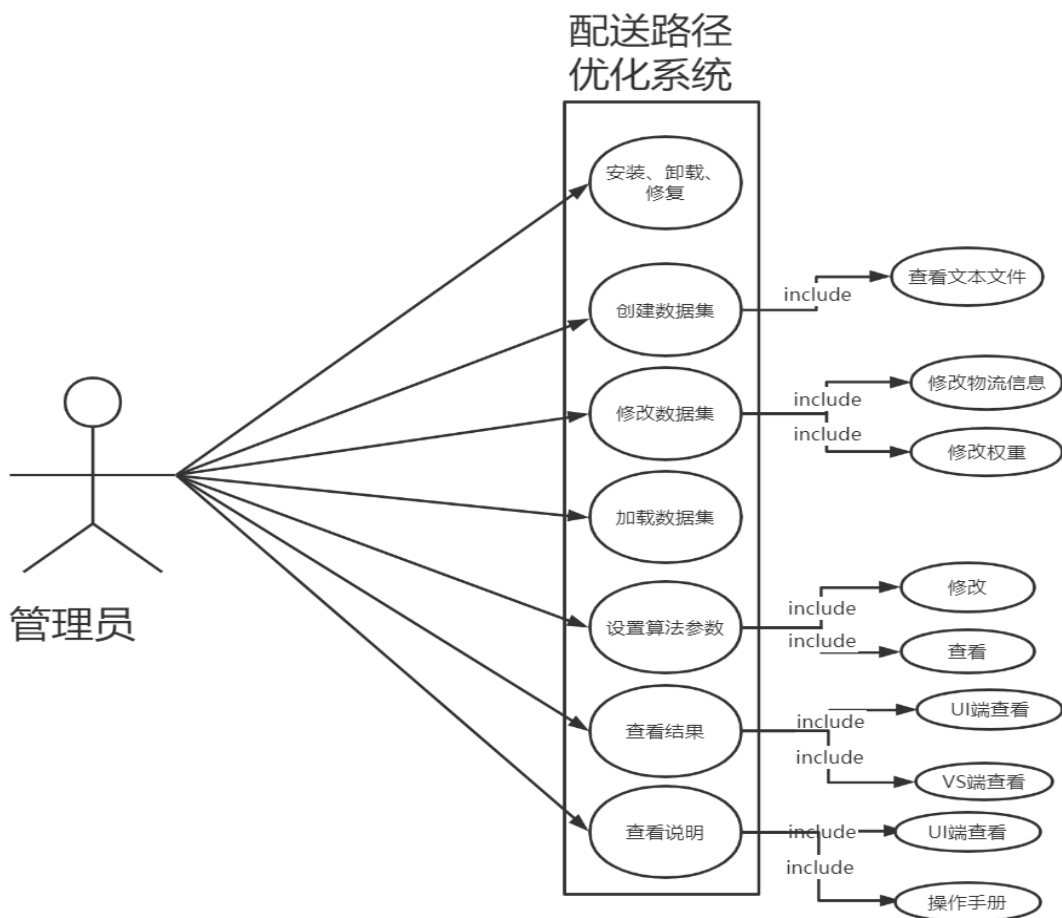


图 3.1.1 管理员用例图

管理员作为系统的最高级用户，拥有全部的功能。系统管理员能够在最高层次上对整个系统的运行进行管理，使系统能够正常完成所有期望功能及在最大程度上解决未知错误。

管理员可以进行如下的操作：

①数据集创建及修改：

用例描述：该用例提供了用户输入和修改的 UI 界面，以及保存输入信息的 txt 文件。

用例名称：数据保存及修改。

前置条件：软件已安装且可以运行软件。

假设条件：假设用户可以完美运行 CVRP 程序。

基本操作流程：1. 到指定目录打开软件。2. 点击创建按钮，输入地图信息后点击确认，系统检测是否输入正确。3. 点击修改按钮对指定文件进行修改保存。

②程序安装、卸载、修复：

用例描述：用户可以通过 setup.exe 对程序进行修复、安装和卸载。

用例名称：程序安装及卸载。

前置条件：可以下载并打开程序安装包。

假设条件：用户已经成功打开安装程序。

基本操作流程：

1. 下载 CVRP 压缩包。
2. 打开压缩包中的 setup.exe。
3. 点击安装、下一步将程序安装到指定目录。
4. 再次运行 setup.exe 删除或者修复程序。

③数据集加载：

用例描述：通过 UI 界面的加载文件按钮将数据集中的数据加载到算法系统当中。

用例名称：数据集加载。

前置条件：软件已安装且可以运行软件，已经创建好数据集。

假设条件：假设用户可以完美打开 CVRP 程序。

基本操作流程：1. 到指定目录打开软件。2. 点击加载文件，指定目录到创建好的文本文件。3. 点击确认。

④设置算法参数：

用例描述：通过 VS 打开软件源代码，对 CVRP.cpp 中的迭代次数变量和种群规模变量进行修改，需要参考用户手册以及其他文档，一般不需要做出改动。

用例名称：设置算法参数。

前置条件：软件已安装且可以运行软件。

假设条件：假设用户可以完美运行 CVRP 程序。

基本操作流程：1. VS 端打开程序源代码。2. 找到变量并修改，保存。3. 导出安装包重新安装。

⑤查看结果：

用例描述：加载完文件后，系统运行路径规划代码，得出最终方案并显示在 UI

界面。

用例名称：查看结果。

前置条件：加载成功且附加信息得到确认。

假设条件：用户已完成加载文件和附加信息确认。

基本操作流程：

1. 打开程序，加载文件并显示加载成功，输入附加信息并确认。
2. 点击展示按钮，系统计算 3-5 分钟。
3. 系统将运行结果显示到 UI 界面。

⑥查看说明

用例描述：在用户首页点击“关于”，查看相关指南和信息。

用例名称：查看说明。

前置条件：成功打开软件。

假设条件：用户以及成功安装并打开。

基本操作流程：在用户首页点击“关于”，查看相关指南和信息。

3.1.2 普通用户用例分析

普通用户用例图如 3.1.2 所示：

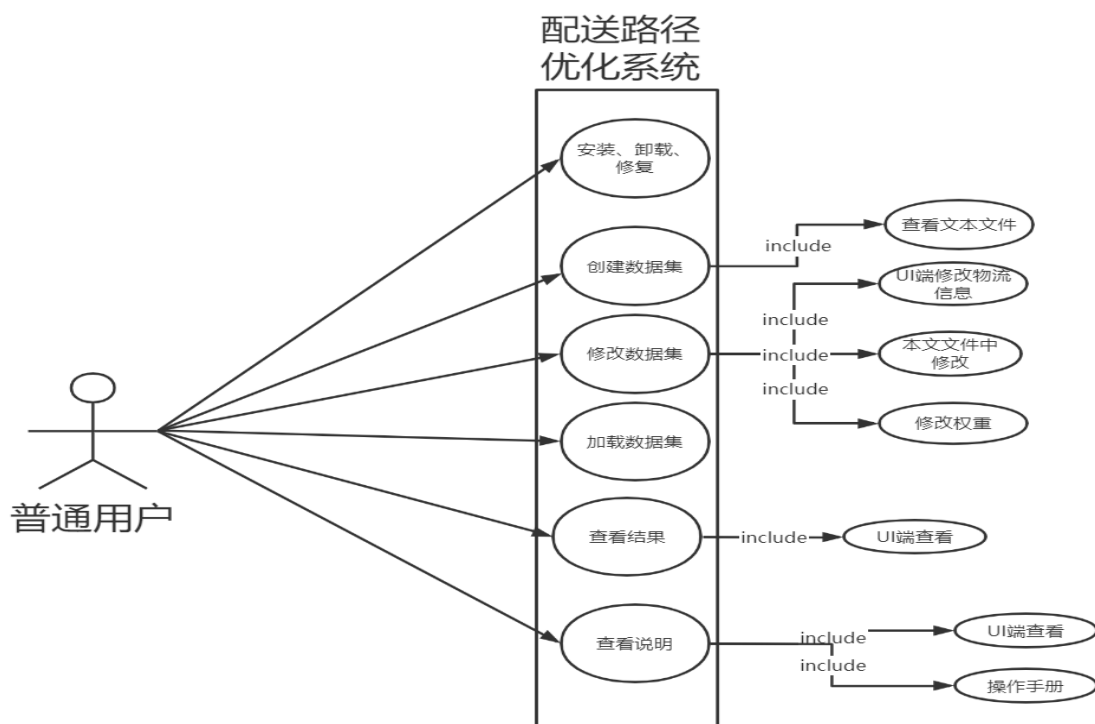


图 3.1.2 普通用户用例图

普通用户是本软件系统的主要面向的对象，系统提供了大部分的功能供用户使用，对用户来说简单而且绰绰有余。
普通用户可以执行以下操作：

①数据集创建及修改：

用例描述：该用例提供了用户输入和修改的 UI 界面，以及保存输入信息的 txt 文件。

用例名称：数据保存及修改。

前置条件：软件已安装且可以运行软件。

假设条件：假设用户可以完美运行 CVRP 程序。

基本操作流程：1. 到指定目录打开软件。2. 点击创建按钮，输入地图信息后点击确认，系统检测是否输入正确。3. 点击修改按钮对指定文件进行修改保存。

②程序安装、卸载、修复：

用例描述：用户可以通过 setup.exe 对程序进行修复、安装和卸载。

用例名称：程序安装及卸载。

前置条件：可以下载并打开程序安装包。

假设条件：用户已经成功打开安装程序。

基本操作流程：

1. 下载 CVRP 压缩包。
2. 打开压缩包中的 setup.exe。
3. 点击安装、下一步将程序安装到指定目录。
4. 再次运行 setup.exe 删除或者修复程序。

③数据集加载：

用例描述：通过 UI 界面的加载文件按钮将数据集中的数据加载到算法系统当中。

用例名称：数据集加载。

前置条件：软件已安装且可以运行软件，已经创建好数据集。

假设条件：假设用户可以完美打开 CVRP 程序。

基本操作流程：1. 到指定目录打开软件。2. 点击加载文件，指定目录到创建好的文本文件。3. 点击确认。

④查看结果：

用例描述：加载完文件后，系统运行路径规划代码，得出最终方案并显示在 UI 界面。

用例名称：查看结果。

前置条件：加载成功且附加信息得到确认。

假设条件：用户已完成加载文件和附加信息确认。

基本操作流程：

1. 打开程序，加载文件并显示加载成功，输入附加信息并确认。
4. 点击展示按钮，系统计算 3-5 分钟。
5. 系统将运行结果显示到 UI 界面。

⑤查看说明

用例描述：在用户首页点击“关于”，查看相关指南和信息。

用例名称：查看说明。

前置条件：成功打开软件。

假设条件：用户以及成功安装并打开。

基本操作流程：在用户首页点击“关于”，查看相关指南和信息。

3.2 非功能性需求

3.2.1 应用兼容性

配送路径优化系统终端目前只可运行在 PC 端，具有可执行版本与编译器运行的版本。对硬件要求不高，windows98 以上的 PC 机都可以完美运行，具有良好的硬件兼容性。

3.2.2 性能需求

配送路径优化系统一般应用于工程前期的规划阶段，对速度要求不高，对精度要求较高，本系统在参数配置上倾向于精度，达到 99%以上的精度，运行速度在 5 分钟左右，基本可以达到企业的性能需求。

3.2.3 可靠性需求

对一般数据集的运算精度可达到 99%以上，软件运行过程无卡顿无错误，软件绿色健康无毒，无需联网便可运行，具有优秀的安全保障。软件大小小于 500M，对计算机存储几乎没有影响。

3.2.4 可用性需求

在电力、计算机正常的情况下保证正常使用。正常使用时间大于全部使用时间的 99%。

3.2.5 出错处理需求

当出错时给用户显示错误提示并给出反馈，给错误处理人员提供可查询的相关信息及代码。保证软件的正常状态。

3.2.6 约束

- (1) 操作系统最好为 windows7/10。
- (2) 使用之前最好阅读操作手册。
- (3) 需要确保安装无误，防火墙不会阻挡。

3.2.7 逆向需求

配送路径优化系统不应处理错误的数据集。

3.2.8 将来可能提出的要求

- (1) 继续优化算法模型，得出更高精度的结果；
- (2) 优化数据结构，提高运行速度，减少内存占用；
- (3) 数据集检测，检测数据集格式的正确性，给出反馈；
- (4) 数据库设计，设计数据库用来对大数据维护和处理；
- (5) APP 设计，提供手机端的终端；

3.2.9 可扩充性

系统需要具体良好的可扩充性，便于配送路径优化系统的部署。可在当前需求基础之上进行扩展，支持更大的数据集和更多的使用场景，支持更多可选择的功能。

3.2.10 可维护性

程序采用面向对象的思想编写，对后期维护和扩展带来很大的便易，如果需要效率上的优越，可移植到更高配置的 PC 机上运行。

3.2.11 可移植性

该软件目前仅支持在 windows 操作系统下运行使用，最好使用 windos7 及以上操作系统。

3.2.12 美观性

要求设计简单的使用界面，本项目使用 QT 设计 UI 界面，操作简单，整洁美观。

3.2.13 易用性

程序符合普通用户使用习惯，操作易上手，对输入过程也会有错误提示。

3.2.14 正确性

软件运行错误不允许出现。

3.2.15 健壮性

应保证在主流计算机流畅运行，软硬件不更新的情况下永久使用。

3.2.16 清晰性

业务流程明确，控制流明确，数据流明确。界面信息清晰，可以正常使用。

3.2.17 安全性

程序数据保存在本地电脑，用户可以对其修改删除，不用担心泄露风险，具有良好的保密性。

3.3 应用对象

本程序应用对象为没有计算机知识的普通用户和有一定计算机基础的管理人员。为了迎合广大用户的需求，程序使用 QT 开发了简洁明了的 UI 界面，只需要简单地输入一些数据就可以得到理想方案，对用户的宽容性较高。程序代码结构清晰，备注完善，对维护人员的后期工作带来便易。

3.4 应用环境

开发环境:

编译环境: QT5, Visual Studio2017, Windows10 操作系统

开发语言: C/C++。

运行环境:

内存: 8G。

CPU: corei7 处理器。

操作系统: windows10。

4. 技术路线及设计方案

4.1 技术路线

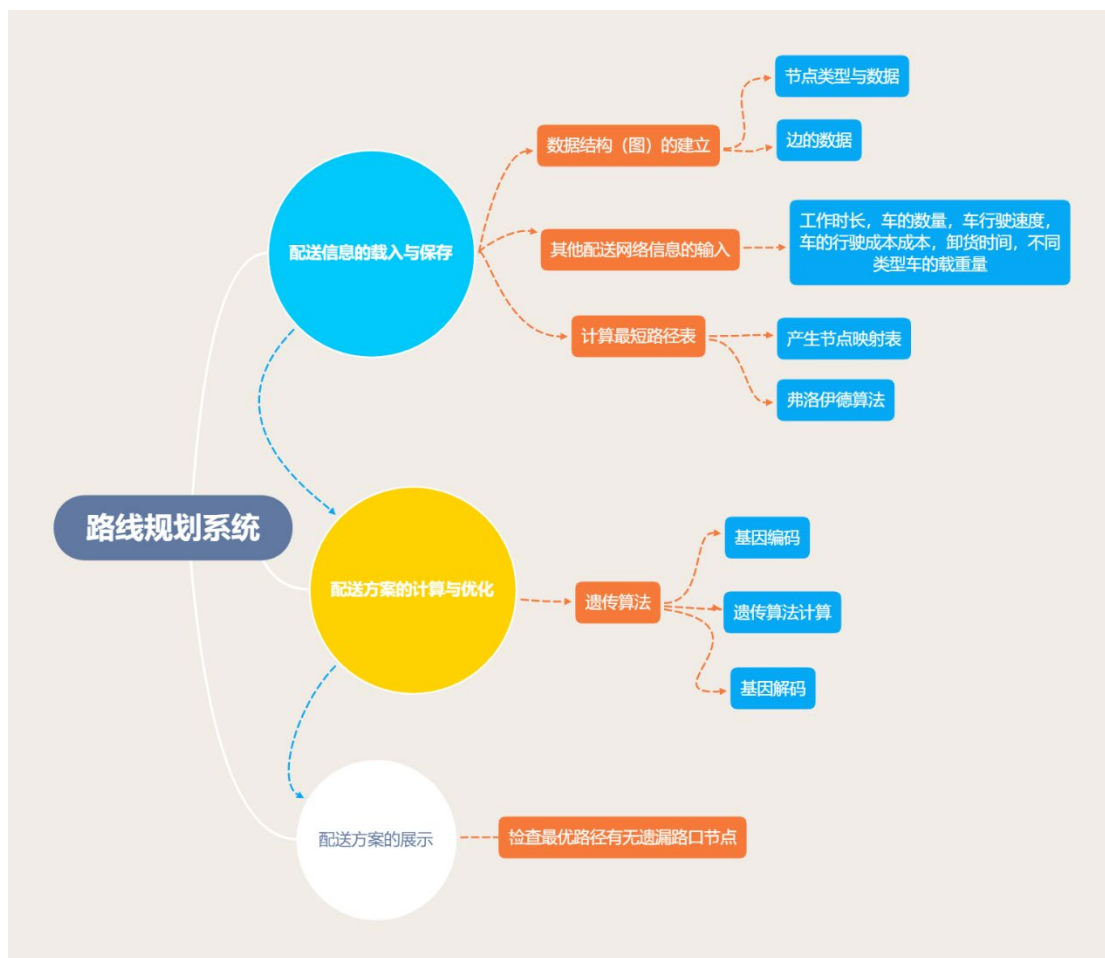


图 4.1.1 开发过程思维导图



图 4.1.2 项目研究路线图

项目关键技术:

1. 弗洛伊德算法:

弗洛伊德算法用来计算图的各点之间的最短距离,得到最短路程表和最短路径表,同时建立节点映射表,将所有的配送点之间的最短距离记录,并排除路口节点对基因的干扰。路口节点会在之后的路径检查中,如果在最优路径上,将会被添加。弗洛伊德算法实现较为简单,故不在后面详细介绍。

2. 遗传算法:

遗传算法 (Genetic Algorithm, GA) 最早是由美国的 John holland 于 20 世纪 70 年代提出,该算法是根据大自然中生物体进化规律而设计提出的。是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型,是一种通过模拟自然进化过程搜索最优解的方法。该算法通过数学的方式,利用计算机仿真运算,将问题的求解过程转换成类似生物进化中的染色体基因的交叉、变异

等过程。在求解较为复杂的组合优化问题时,相对一些常规的优化算法,通常能够较快地获得较好的优化结果。遗传算法,是本项目的核心算法,将在之后的算法设计中会被详细介绍。

3. 数据结构:

本项目采用的数据结构为图,采用邻接表来储存图的相关信息。

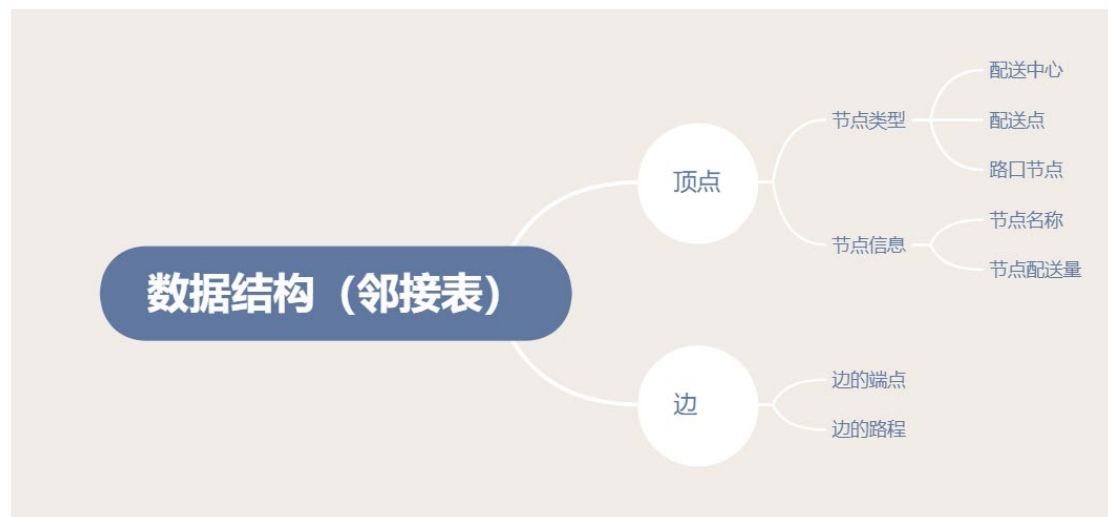


图 4.1.3 数据结构图

4. 算法模型

算法模型也是本项目的核心之一,经过我们项目小组的反复讨论,我们将在后面的模型设计一栏中,详细介绍算法模型。

4.2 模型设计

基于本项目的实际情况，我们提出了如下的数学模型：

由单一起点出发，配送多个终点后再回到起点，根据车辆数量，承载限制，不同车辆服务成本、运行里程限制等条件选择最优运输路径，使成本最小化，配送订单最大化，满载率最大化（如由一个配送中心向各个销售点配送货物，通过算法确定配送中心每辆车的配送方案，包括配送至哪个客户，配送量，下一个配送目的地）。

根据上诉要求，我们提出了以下模型：

设配送中心有 K 种车辆，每种车型有 m_k 辆车 ($k=1, 2, 3, \dots, K$)，每种车型的载重量为 h_k ，每种车型服务成本费用为 o_k ，共有 N 个配送点， V_n 表示配送点，配送点需求量为 q_i ，需求点 i 到 j 的运距为 d_{ij} ， s 表示为车辆的编号， X_{ijs} 表示车辆 s 是否从 i 跑到 j ，线路编号 $r=1 \dots n$ ， k_r 表示 r 线路的运输车编号， F_a 表示第 a 种车型的使用数量， C_r 表示第 r 条路径上的配送点数， Y_{ik} 表示配送点 i 是否被车辆 k 服务， n_a 表示第 a 辆车配送点数， T 表示配送限制时间， t 表示卸货时间， $Signs$ 表示车辆 s 是否参与配送， D 表示车辆最大行驶距离，则可建立如下物流配送路径优化问题的数学模型：

$$\min F = wc * \min C + wd * \min d + wk * \min k + wl * (1 - \max L) \quad (1)$$

$$\min C = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^K o_k x_{ijk} d_{ij} \text{sign}(n_k) \quad (2)$$

$$\min D = \sum_{s=1}^k \sum_{i=1}^N \sum_{j=1}^N x_{ijs} d_{ij} \text{sign}_s \quad (3)$$

$$\min N = \sum_{k=1}^K \text{sign}(n_k) \quad (4)$$

$$\max L = \sum_{s=1}^k \frac{Q_s}{h_K} \quad (5)$$

$$\sum_{i=0}^n \sum_{j=0}^n x_{ijk} d_{ij} \text{sign}(n_k) \leq D_0 \quad (6)$$

$$R_r = \{r | r \in \{1, 2, \dots, n\}\}, \forall a, b \in r \wedge a \neq b, R_a \cap R_b = \emptyset \quad (7)$$

$$a_r = V_0 \quad b_r = V_0 \quad a_r, b_r \text{ 分别是第 } r \text{ 条路线的起点和终点} \quad (8)$$

$$\frac{1}{S} \sum_{i=0}^n \sum_{j=0}^n x_{ijk} d_{ij} \text{sign}(n_k) + \sum_{i=0}^n \sum_{j=0}^n x_{ijk} \text{sign}(n_k) t \leq T \quad (9)$$

$$F_a \leq M_a, \quad a \in \{1, 2, 3, \dots, K\} \quad (10)$$

$$Q_r \leq h_{k_r} \quad (11)$$

$$\text{sign}(n_k) = \begin{cases} 1 & n_k \geq 1 \\ 0 & \text{其他} \end{cases} \quad (12)$$

$$\sum_{k=1}^K n_k = n \quad (13)$$

上述模型中，(1) 式为目标函数，根据四个不同的权重来调整在服务成本，路程，载重率和车的数量四者之间的关系；(2) 式计算服务成本；(3) 式计算最短路程；(4) 式计算方案所用车的数量；(5) 式计算载重率；(6) 每条路线的长度不能超过车辆巡回里程限制；(7) 式 2 限制每个车只能服务一条路线且每个配送点只能由一辆车配送；(8) 式限制每条路线的起始点和终点是配送中心；(9) 式表示每条线路的运行时间不能超过配送史称；(10) 式表示车的使用数量不能超过车的总数量；(11) 式表示每条路线配送货物之和不能超过车的承载量；

(12) 式表示第 k 辆车服务的配送点数大于等于 0 时，说明该车参与了配送；

(13) 式表示每个配送点都能得到配送。

4.3 算法设计

针对物流配送路径优化问题的特点，构造了求解该问题的遗传算法。

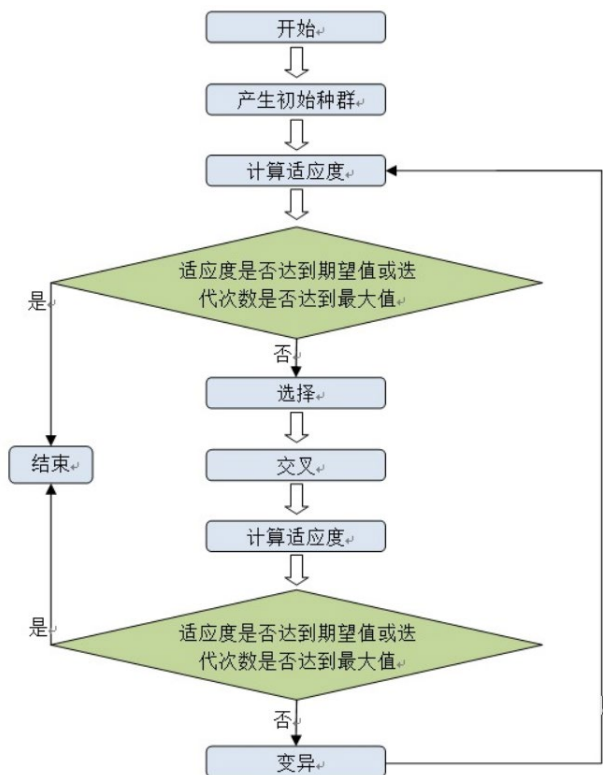


图 4.3.1 遗传算法基本流程图

4.3.1 染色体编码

除配送中心外，要配送的配送点数目为 N ，从 1 到 N 表示货物需求点编号，0 表示中心仓库编号。染色体种的基因位置的表示配送点或者配送中心的编号。染色体整个基因的次序就表示车辆访问配送点的次序，由于参与配送的车辆数目是不确定的，如果每一个配送点都需要派一辆车单独运送，那么染色体的最大的可能长度为 $2N+1$ ，因此染色体的长度就为 $2N+1$ 。

在此基础上又设立两个辅助数组，第一个辅助数组中的每一个元素用于储存每条子路径需要的货物总量，并按照染色体中表示的子路径的访问顺序依次存放。第二个辅助数组用来存放每条子路径派出的车的最大载重量，并按照染色体中表示的子路径的访问顺序依次存放，最后一个元素用来存放参与配送的车辆数目。

应用此编码方式有两个优点：第一，可以方便的得到某个配送点是由哪辆车服务的，每辆车服务的需求点数目以及具体的配送点情况，所需要的车辆数量等信息，不会造成重要信息的丢失。第二，在执行遗传算子的时候，仅对染色体本身进行遗传操作，其对应的辅助数组做相应的调整即可，这对增强算法的收敛性和缩短求解实践具有重要意义。例如：共有 9 个配送点和一个配送中心编号为 0 1 2 3 4 5 6 7 8 9

染色体 1: 0 15 13 7 4 11 3 9 17 12 14 16 2 6 5 1 8 10 18

所以三条路径为: 0 7 4 0 | 0 3 9 0 | 0 2 6 5 1 8 0

4.3.2 初始种群的确定

在标准的遗传算法求解中，初始种群的生成方式一般是基于随机生成的方式，所以本项目采用初始种群的方式是随机生成。

4.3.3 适应度函数的设定

对于某个个体所对应的配送路径方案，要判定其优劣，一是要看其是否满足配送的约束条件；二是要计算其目标函数值（即各条配送路径的长度之和）。本文根据配送路径优化问题的特点所确定的编码方法，隐含能够满足每个需求点都得到配送服务及每个需求点仅由一辆汽车配送的约束条件，但不能保证满足每条路径上各需求点需求量之和不超过汽车载重量及每条配送路线的长度不超过汽车一次配送的最大行驶距离的约束条件。

为此，对每个个体所对应的配送路径方案，要对各条路径逐一进行判断，看其是否满足上述两个约束条件，若不满足，则将该条路径定为不可行路径，最后计算其目标函数值。

对于某个个体 j ，设其对应的配送路径方案的不可行路径数为 M_j ($M_j=0$ 表示该个体对应一个可行解)，其目标函数值为 Z_j ，则该个体的适应度 F_j 可用下式表示：

$$F_j = \frac{1}{Z_j + G * M_j}$$

式中， G 为对每条不可行路径的惩罚权重，可根据目标函数的取值范围取一个相对较大的正数。

4.3.4 遗传算法的设计

(1) 选择

本项目采用锦标赛选择法和精英保留策略相结合的选择策略：将每代种群的所有染色体按照适应度的值从大到小排序，选择部分适应度最高的个体，剩下的个体通过锦标赛选择法选择，然后组合成新的种群，将新的种群按照适应度从大到小排列，将适应度最高的个体保存，然后进行交叉和变异操作，生成新的种群，如果新的种群中最高适应度小于旧种群中最高适应度，则用旧种群中适应度最高的个体代替新种群中适应度最低的个体。

(2) 交叉

本项目采用的是基于顺序的交叉的方式，即根据交叉概率，对选择出来的每条染色体中表示车辆路径部分进行顺序的炒作。例如：

1) 第一步，随机选择一对染色体（父代）中几个基因，位置可不连续，但两染色体被选位置相同：

Parent 1 : 1 2 3 4 5 6 7 8 9

2) 第二步, 先在父代 2 中找到父代 1 被选中基因的位置, 再用父代 2 中其余的基因生成子代, 并保证位置对应, 将父代 1 中被选择的基因按顺序放入子代剩余位置中:

Parent 1 : 1 2 3 4 5 6 7 8 9

Parent 2 : 5 4 6 3 1 9 2 7 8

Children : 2 4 5 3 1 6 9 7 8

(3) 变异

本项目的变异操作较为简单, 根据变异概率, 随机产生一个数判断是否小于变异概率, 如果小于则连续进行三次变异, 随机产生三对不重复的点, 进行对内交换。

4.3.5 交叉算子和变异算子的改进

由于交叉概率和变异概率是进行交叉和变异操作的重要参数, 对遗传算法的搜索能力有很大影响, 而基本遗传算法的交叉概率和变异概率是设定的固定值, 容易陷入局部最优解。所以本项目采用自适应的交叉算子和变异算子的改进方法。

交叉概率和变异概率变化公式如下:

$$P_c = \begin{cases} P_{c1} - \frac{(P_{c1} - P_{c2})(f' - f_{avg})}{(f_{max} - f_{avg})[1 + \exp(-t*S/T*M)]}, & f' \geq f_{avg} \\ P_{c1} [1 + \exp(-t*S/T*M)], & f' < f_{avg} \end{cases} \quad (14)$$

$$P_m = \begin{cases} P_{m1} - \frac{(P_{m1} - P_{m2})(f_{max} - f)}{(f_{max} - f_{avg})[1 + \exp(t*S/T*M)]}, & f \geq f_{avg} \\ P_{m1} [1 + \exp(t*S/T*M)], & f < f_{avg} \end{cases} \quad (15)$$

其中: (14) 式为交叉的概率变化公式; (15) 式为变异的概率变化公式; f_{max} 为每代种群最大的适应值, f_{avg} 为每代种群的平均适应值, f' 为要交叉的个体中比较大的适应度, f 为要变异个体的适应值, t 为当前进化代数, S 为进化过程中最优解没改变的个体数目, T 为最大终止代数, M 为种群大小, 其中 $P_{c1} > P_{c2}$, $P_{m1} > P_{m2}$, 取 (0, 1) 区间的值, 可以在优化过程中调整。

进化初期, 种群中的个体之间差异比较大, $f_{max} - f_{avg}$ 的值也比较大, 此时进化代数 t 和进化过程中个体未改变数目 S 比较小, 所以交叉概率 P_c 比较大, 种群收敛速度较快。进化后期, 由于种群中个体之间的差异变小, 个体之间相似度变高, 收敛速度变慢, 有陷入局部最优的趋势, $f_{max} - f_{avg}$ 的值变小, 进化代数 t 和进化过程中个体未改变数目 S 变大, 为了提高收敛效率, 交叉概率 P_c 变小, 变异概率 P_m 增加, 提高了种群个体的多样性, 充分显示了变异算子的局部搜索能力。经过改进的遗传算法在进化的过程中根据适应度大小、进化代数 t 和进化过程中个体未改变数目 S 自适应调整交叉和变异概率, 不仅能保证交叉操作的全局搜索能力, 并且能充分发挥变异操作的局部搜索能力。

5. 项目管理与业务模式

5.1 项目计划



图 5.1.1 项目计划图

5.2 项目估算

项目估算是指根据项目的开发内容、开发工具、开发人员等因素对需求调研、程序设计、编码、测试等整个开发过程所花费的时间及工作量做的预测。本项目组使用 Delphi 法对工作量进行估算。Delphi 估算法是一种专家估算技术，在没有历史数据的情况下，这种方式适用于评定过去与将来新技术与特定程序之间的差别。由于开发团队的 3 位成员都有一定的项目开发经验，因此由 3 位开发团队成员组成专家小组，参与 Delphi 估算，产品负责人主持估算。

本项目组的 Delphi 估算步骤如下所示：

- (1) 产品负责人召集小组会议，向各专家小组成员提供产品待办列表和估算表格，并对估算内容进行介绍。
- (2) 每位成员独立进行估算，匿名填写估算表格，提交给产品负责人。
- (3) 产品负责人用估算汇总表总结各估算人员的估算结果。
- (4) 各成员针对差异进行讨论，提出各自的估算理由，互相说服对方，但不涉及具体的估算值。讨论时间限制在 5-10 分钟之内，避免无限制的讨论。
- (5) 差异讨论之后，开始下一轮独立估算，专家小组成员根据在讨论中获得的新信息，以及参考别人的意见，各自对自己的估算进行调整，重复 2-5 步骤，直至达到一个最低和最高估算的一致，获得一个多数专家达成共识的软件规模或工作量估算。

5.3 角色分配与职责

姓名	角色	职责
小郑	项目经理	1. 项目整体规划；2. 项目策划设计；3. 核心技术及方案实现；4. 项目沟通管理；5. 项目人员调整；6. 项目进度安排；7. 项目风险管理；
小刘	算法分析师	1. 项目架构设计；2. 技术方案设定；3. 算法模型的设计及实现；4. 核心技术及方案；5. 技术问题处理；6. GUI 设计；
小顾	测试工程师	1. 项目质量管理；2. 系统的详细测试；3. 核心技术及方案实现；4. 功能模块对接；

小梁	需求分析师	1. 视频拍摄；2. 研究调查分析；3. 核心技术及方案实现；4. 产品的文案和推广
----	-------	--------------------------------------------

图 5.3.1 项目职责分配表

5.4 开发管理

参照瀑布模型开发管理方法，结合项目和团队实际情况，本项目组的开发流程如图 5.4.1 所示：

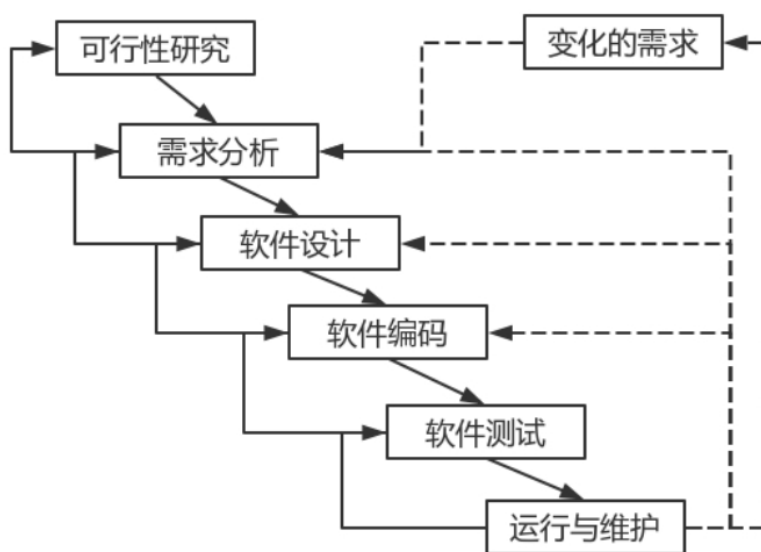


图 5.4.1 开发流程

开发团队在执行每个开发阶段需要交付符合一定标准的、完整的文档及资料。并举行一次大型的项目会议，讨论项目进度以及现阶段遇到的困难等问题。由于开发周期较短，结合团队实际情况，我们采用每日站立会议来汇报工作进展和面临的障碍。

在工作日的每一天早晨，团队定时举行不超过 15 分钟的每日站立会议，所有团队成员都必须参加。开发团队的每一位成员都必须轮流汇报以下 3 个问题，即：

- 昨天我完成了什么工作？
- 今天我打算做什么？
- 我的工作遇到了什么困难和障碍？

通过这 3 个问题，团队成员之间相互熟悉工作内容，充分了解项目进度，增强了成员之间的交流沟通。项目经理除了倾听开发团队成员的发言外，同时还会把障碍记录下来，以备会议后帮助开发团队成员排除障碍。团队成员之间在会后也可以单独会谈，互相帮助解决工作中的困难。

5.5 项目监控

5.5.1 项目评审

在项目研发过程中为了保证研发进度及研发产品的质量，会安排大量的线上评审活动。本系统在开发过程中通过项目评审作为项目管理的一部分来对研发过程和产品质量的保证。评审记录如下图所示：

工作产品	评审方式	评审时间	评审人员
可行性分析	正式	2020 年 1 月 20 日	全体成员及指导老师
需求说明书	正式	2020 年 2 月 1 日	全体成员及指导老师
资料查阅、收集	正式	2020 年 2 月 7 日	全体成员及指导老师
概要设计说明书	正式	2020 年 2 月 17 日	全体成员及指导老师
详细设计说明书	正式	2020 年 3 月 6 日	全体成员及指导老师

图 5.5.1.1 评审记录

任务名称	开始时间	结束时间	参与人员	工作量 (人/天)	工作成果
可行性分析	2020-1-15	2020-1-20	全体成员	25	可行性分析报告、 团队分工
需求分析阶段	2020-1-21	2020-2-1	全体成员	25	项目管理计划、需求规格说明书
技术准备阶段	2020-1-31	2020-2-7	全体成员	25	编码规范、 技术说明。
概要设计阶段	2020-2-6	2020-2-17	全体成员	100	项目开发计划、 概要设计说明书

详细计划阶段	2020-2-18	2020-3-6	全体成员	100	项目开发详细计划、质量保证计划、测试计划
系统设计阶段	2020-3-5	2020-3-11	全体成员	25	概要设计、数据库设计、详细设计、测试计划。
系统编码阶段	2020-3-10	2020-3-20	全体成员	20	源代码、可执行与安装文件。
系统功能集成阶段	2020-3-19	2020-3-24	全体成员	20	源代码、功能集成
系统测试阶段	2020-3-23	2020-3-31	全体成员	25	功能测试、集成测试、系统测试、非功能测试、测试文档、用户操作手册、安装说明。
系统修正阶段	2020-4-1	2020-4-7	全体成员	30	源代码定稿、安装可执行与文件定稿、用户操作手册定稿。

图 5.5.1.2 进度管理图

5.5.2 项目过程与质量管理

(1) 项目过程管理

项目组制定项目开发计划，建立人员组织，并进行人员分配。根据项目开发生命周期启动项目。召开项目会议，一周一次大会，每天一次小会，并建立会议文档，保证项目过程出现问题的解决。项目经理扮演项目监督的角色，对项目生命周期中的正常运行情况进行监督并对出现的问题进行处理。

(2) 项目质量管理

客户关系由需求分析师进行跟踪，对于客户的需求内容以及文化推广方面的考虑，客户经理需要对项目进行过程中是否围绕需求进行跟踪。方案的批准、纠正、缺陷修复等措施需要明确的处理。项目范围的保证，时刻根据需求，把项目限定在准确的范围内进行开发。

5.5.3 项目沟通与风险管理

(1) 项目沟通

沟通成员组成：项目经理、算法工程师、测试工程师，需求分析师。制定明确的沟通计划与风险管理方案。

沟通通过会议正式决议，以及即时通信工具、电话、邮件、面谈作为临时讨论来进行沟通。

沟通时长根据会议大小而定，一次大型会议，沟通时长为一个小时，小型会议的沟通时长为半小时。

(2) 风险管理

风险需要限定在可控的范围内。风险的认定需建立在项目组成员认可的基础上。风险的监控由项目经理主要负责。技术风险需要技术经理进行认定。

5.6 成本分析

(1) 成本构成

如图 5.6.1 所示，基于遗传算法的路径规划系统共的成本包括研发成本、推广成本和维护成本。研发成本包括基于遗传算法的路径规划系统共研发过程中所产生的一切费用，如调研费用、开发费用、测试费用、人工费用、水电费等。推广成本包括基于遗传算法的路径规划系统共推广过程中所产生的一切费用，如广告宣传费用。维护成本包括基于遗传算法的路径规划系统共维护过程中所产生的费用。



图 5.6.1 成本构成

(2) 研发成本分析

基于遗传算法的路径规划系统的研发成本分为直接成本和间接成本。直接成本为项目组在项目开发过程中的直接开销，间接成本为项目开发过程中与项目无关的一些开销费用如水电费、场地租借费等。表列出了项目组在项目制作过程中产生的直接成本和间接成本。

研发成本类型	费用名称	计算方法	费用(元)
直接成本	人员工资	$3(\text{人}) * 5000 * 1.3(\text{月}) + 1(\text{人}) * 6000 * 1.3(\text{月})$	27300
	设备折旧费	$4(\text{电脑}) * 8000(\text{元}) * 5\%(\text{折旧率}) * 1.3(\text{月}) + 4(\text{Android 手机}) * 3000(\text{元}) * 1.3(\text{月}) * 5\%(\text{折旧率为 } 5\% \text{ 每月})$	2860
	差旅费、会议费	每人差旅补贴 200 元	800
	其他办公杂费		1200
	项目测试费用		1500
	资料费		800

	调研成本	1 (次) * 1200 (元)	1200
间接成本	水费	1.85 (元/立方米) * 2 (吨)	3.7
	电费	0.63 (元/度) * 300 (度)	189
	其他补贴及津贴	4 (人) * 60 (元/人) * 1.3 (月)	312
合计			36164.7

图 5.6.2 成本分析表

(2) 推广成本与维护分析

基于遗传算法的配送路线优化系统将从 2020 年开始被使用，该系统 2020-2020 年的推广成本和维护成本预测如下表所示。

年份	推广成本	维护成本	合计 (万元)
2020	40.0	10.0	50.0
2021	60.0	30.0	90.0
2022	100.0	40.0	140.0

图 5.6.3 2018-2020 年推广及维护成本

测算依据如下：

推广方式：我们团队设计了一款轻量级的软件，可以离线操作，并且使用了 Visual Studio 2017 来进行编译，所以只能在 windows 操作系统的终端中使用。软件的推广实行按季度或者年度收费。

维护方式：及时发布最新版本的软件。

(1) 预测 2020 年基于遗传算法的配送路线优化系统推广成本为 40 万元。预计将有 5 万名配送人员使用，由于是软件更新，所以一年的维护成本约为 10 万元。

(2) 预测 2021 年基于遗传算法的配送路线优化系统推广成本为 60 万元，预计有近 10 万配送人员使用，维护成本近 20 万元。

(3) 预测 2022 年基于遗传算法的配送路线优化系统成本为 100 万元。2022 年开始预计有 20 万左右配送人员使用该系统，维护总成本为 40 万元。

5.7 市场与竞争分析

5.7.1 市场分析

（一）物流行业规模进一步扩大：

随着中国经济的持续稳定发展，经济体量越来越大，随之而来的便是物流规模的进一步扩大，据物流权威人士估计，2030 年，中国将成为全球贸易巨人，中国与主要经济体、新兴经济体、发展中国家的贸易会进一步提升，中国的国际物流规模会有更大扩张。2020、2030 年电子商务产生的日快递量将分别突破 1 亿件和 2 亿件。

（二）科技化、专业化趋势进一步加强：

信息网络领域的技术突破，大型高速船舶、新能源汽车、无人驾驶、物联网、下一代信息网络等将在物流领域得到广泛应用，互联网、大数据、云计算、人工智能等将与物流业深度融合，这些都对物流业升级具有重大促进作用。未来，物流技术创新将显著反映出快速便利、自动化、信息化、网络化等时代性特点。

（三）物流配送的效率很不理想：

根据国家发展改革委、国家统计局等部门联合发布的数据显示，2018 年全国社会物流总额 283.1 万亿元，按可比价格计算，同比增长 6.4%，增速比上年同期回落 0.2 个百分点。其中，运输费用 6.9 万亿元，增长 6.5%，增速比上年同期下降 4.3 个百分点，运输费用与 GDP 的比率为 7.7%，比上年同期下降 0.3 个百分点。保管费用 4.6 万亿元，增长 13.8%，增速比上年同期提高 7.1 个百分点，保管费用与 GDP 的比率为 5.1%，比上年同期提高 0.4 个百分点。管理费用 1.8 万亿元，增长 13.5%，增速比上年同期提高 5.1 个百分点，管理费用与 GDP 的比率为 2%，比上年同期提高 0.1 个百分点。社会物流总费用占 GDP 比率虽然连续六年持续下降，但依旧偏高，阻碍中国经济全球竞争力。

分析结果

社会物流费用的偏高，是因为现阶段物流配送的成本偏高，而本项目迎合为来物流的科技化专业化趋势，助力于企业扩大物流规模，有助于体改物流配送的效率，在市场上是有大有可为的。

5.7.2 竞争分析

本项目与市面上的软件相比的优势：

（1）首先，市面上的软件只是单纯的寻找最短路径，并没有为企业考虑成本，工作时间，没有考虑载重率的各种复杂的因数，所以导致现在物流企业的仓储保管费用逐年升高；本项目采用智能算法，可以灵活根据车辆载重率，工作时间和配送中心的车辆进行路线规划，有效提高配送的效率。

(2) 其次，软件是轻量级的软件，装卸载方便，不用联网，不用担心企业和个人信息泄露。

5.7.3 SWOT 分析

SWOT 分析法，即态势分析法，就是将与研究对象密切相关的各种主要内部优势、劣势和外部的机会和威胁等，通过调查列举出来，并依照矩阵形式排列，然后用系统分析的思想，把各种因素相互匹配起来加以分析，从中得出一系列相应的结论，而结论通常带有一定的决策性。

运用这种方法，可以对研究对象所处的情景进行全面、系统、准确的研究，从而根据研究结果制定相应的发展战略、计划以及对策等。SWOT 分析法常常被用于制定集团发展战略和分析竞争对手情况，在战略分析中，它是最常用的方法之一。

市场与竞争分析能够让产品清楚的认识外界因素对项目开发的影响，而利用 SWOT 分析则是将项目和外界相联系，如下图所示，从外部环境整体考虑，更具有实际意义。



图 5.7.3 SWOT 分析

1. 优势 (Strength)

- 1) 有着简洁的 GUI 交互界面，人机友好度高。
- 2) 采用遗传算法，可以根据操作人选择的权重，和输入的配送网络信息和其他配送的信息得出符合操作人预期的合理方案。
- 3) 支持地图载入与修改，不需要操作人每次繁琐的输入地图信息。
- 4) 采用 C/C++ 语言编写，运行速度较快。

2. 弱点 (Weakness)

1) 系统缺陷

目前产品处在初创期，系统还不完善，部分功能仍然欠缺，且数据量较小，分析画像不够精细。

2) 市场

缺乏市场知识与经验，实际应用中的效果具有不确定性，产品可能不能满足

用户的多样化需求

3. 机会（Opportunity）

1) 市场时机

中国物流行业规模的进一步扩大，物流的智能化，科技化发展是未来智慧物流发展的趋势。

3) 社会现状

低效率的配送方式，导致企业仓储成本在物流成本中的比重偏高，阻碍了物流行业的发展，迫切需要合理的规划物流配送方案。

4. 威胁（Threat）

1) 现有的配送路线优化软件

现有市场上有几款类似的路线规划 APP，例如美团和饿了么对送餐员的路线规划。

2) 潜在危险

其他类似的项目，但还没有投放市场；有可能在我们项目投放市场前，迅速形成品牌效应，挤压我们项目的发展空间。

3) 市场占有率

目前产品刚刚起步，市场占有率低，没有形成规模和品牌效应。

5.8 风险控制

5.8.1 风险控制体系

风险指的是相对于预期目标而言，产品遭受损失的不确定性，这种潜在损失在实际中是必定存在的

风险控制是指风险管理者采取各种措施和方法，消灭或减少风险事件发生的各种可能性，或风险控制者减少风险事件发生时造成的损失。

风险控制的四种基本方法是：风险回避、损失控制、风险转移和风险保留。

我们在研发、推广和维护用户行为追踪程序的过程中，肯定会遇到各类问题，怎么能有效进行分析管理，实现在问题发现之前即能使之消灭在萌芽状态？这是本项目组所要解决的一个重要事项。

由此，用户行为追踪程序研发、推广和维护过程中，本项目组制定了一套风险管理及内控体系并不断完善，以保证项目能顺利成功的运行。

风险控制体系详情如下：

（1）项目组一级的年度计划，必需包含风险管理计划；每个项目的开发计划必需包括风险管理计划。

(2) 在制定年度计划或项目计划的同时，必需进行风险识别、风险分析以及风险策划，针对首要风险明确风险责任，确定风险应对策略，制定风险应对行动计划

(3) 项目组应建立风险跟踪制度，跟踪风险和风险管理计划，定期（或事件驱动）验证风险管理活动相对于风险管理计划的符合性。

(4) 在每周或每月的例会上应报告风险。

(5) 在每月或每季的里程碑会议上应评审风险。

(6) 每个项目组应重视经验积累和共享，建立并维护风险数据库。

5.8.2 财务风险及应对措施

财务风险是指财务结构不合理、融资不当使企业可能丧失偿债能力而导致投资者预期收益下降的风险。财务风险是企业财务管理过程中必须面对的一个现实问题，财务风险是客观存在的，管理者对财务风险只有采取有效措施来降低风险，而不可能完全消除风险。

财务风险控制则是指在财务管理过程中，利用有关信息和特定手段，对企业财务活动施加影响或调节，以便实现计划所规定的财务目标，回避风险的发生。财务风险控制的方法主要有：

①防护性控制：在财务活动发生前，制定一系列制度和规定，把可能产生的差异予以排除的一种控制。

②前馈性控制：通过对实际财务系统运行的监视，运用科学的方法预测可能出现的偏差，采取一定措施，使差异得以消除。

③反馈控制：在认真分析的基础上，发现实际与计划之间的差异，确定差异产生的原因，采取切实有效的措施，调整实际财务活动或调整财务计划，使差异得以消除或避免今后出现类似差异。

在我们的项目中，财务风险产生的可能原因如下：

1) 产品的推广是一个产品前期成长的必经过程，其周期较长，耗用资金和时间成本较大，面临的财务风险较大。

2) 资本回收率影响本产品的盈利状况，从而影响本产品的资产结构，若运营不善将会导致本产品处于亏损状态，会增加产品的借入资金，提高了借入资产比例，这会提高财务风险。

3) 市场利率的变化会增加借入资本利息差额的不确定性，这可能会提高借入资金的成本。

为应对财务风险，我们可采取如下对策：

1) 实行严格的资金借贷和运用审批制度，根据产品发展情况和资金市场变化，调节结构。

2) 使投资项目尽快产生收益，提高资产盈利能力，降低投资风险。

3) 加强对业务收入、业务支出、日常现金等的管理，在保持较高的流动性基础上 减少资金占用，为产品扩大投资提供现金流。

4) 加大资本运营的力度，构建和拓宽畅谈的融资渠道，为产品资金供应建立稳固的渠道，为产品的发展不断输入资金。

5) 建立相应的风险预警机制，加强内部管理，严格规章制度，把可能发生的损失降到最低程度。

6) 为避免产品在发生意外及其他不可抗拒因素给企业带来损失，将在财务预算中拨出专款，购买各种保险以规避可能遇到的风险。

5.8.3 技术风险及应对措施

本项目的技术风险主要表现在产品的升级过程中，由于软硬件发展速度较快，而软件使用时可能出现不可预料的情况，或者软件达不到预期的效果，从而影响了产品的竞争力和经济效果

技术风险产生的可能原因：

1) 软硬件发展速度较快，导致产品使用时可能出现不可预料的情况，或者软件 达不到预期的效果；

2) 客户的需求超过技术人员的技术范畴，导致产品无法及时做出符合客户需求 要求升级。

为应对技术风险，我们可采取如下对策：

1) 积极请教技术方面有丰富经验的指导老师，做好面对问题迎难而上解决问题的心理准备，整个团队保持对项目开发的乐观心态。

2) 鼓励技术人员积极相应客户的需求，努力学习更新的技能以应对多变的市场需求。

6. 技术可行分析

6.1 概述

随着生活中不断出现对优化问题的求解需求，以及计算机与数学领域的不断的发展与成熟，产生了许多算法模型与算法实现的方法，为本次的系统开发提供了良好的开发基础。本次的系统开发过程中采用目前成熟与常见的算法与开发语言，为项目的效率与求解可靠性提供了技术上的保障。此外在界面设计上使用的工具，也是目前比较流行的框架。

本次开发使用的 c/c++语言作为开发语言使用，作为算法实现的语言，底层效率较高。

算法模型上选用了遗传算法作为主体框架，在其中加入了动态自适应的改进，避免陷入局部最优的解的情况。

界面设计工具上选用了 QT 开发框架，以提升算法使用的便捷与快速。

我们项目所对应的物流异构车辆调度问题，在日常物流中是一个常见问题，而遗传算法本身作为一种经典的智能算法，广泛用于车间调度中的效果是十分丰硕的。以遗传算法作为主体算法框架，对于项目中的问题进行求解是十分可靠的。因此，本系统设计与开发在技术上具备可行性。

6.2 技术不足风险

6.2.1 面对节点数量大的稠密图，求解时间较长

遗传算法本身因为存在染色体编码的需求，因此在当节点数过多的时候，会出现染色体编码会变得很长、组合方案更多的情况，导致求解的时间变得很长。这是遗传算法本身存在的必然问题，而且在稠密图上求解优化问题本身也是一个无法短时间立即求解的情况，即使可以很快求解也无法保证是否为此问题的最优解。考虑到对最优解的保证，因此无法单从算法的参数上进行调整。

另一方面，在对项目问题进行解析的时候，考虑到问题针对的是一片区域的物流配送问题，而以配送中心为中心的区域划分的情况下，不会或很少会出现节点数路径数过多的情况，因此认为选择遗传算法是可以保证对问题的有效求解的，所以没有去过多考虑节点数量大的稠密图的情况。

规避方案：可以从两方面上进行规避，一方面上是从程序方面，对程序结构上进行优化，改用并行式的运行方式，提升程序整体上的运行的效率，从而提升对于节点数多的稠密图问题的求解速度。另一方面，可以从区域划分上进行入手，可以考虑在对节点数过大配送区域进行重新划分，划分为多个更小的区域，从而将稠密图进行分解，然后依次对更小的区域进行求解，避免了稠密图的情况，在一定程度上减小求解时间提高了运行效率。

6.2.3 对于非法输入缺乏限制，导致程序出错

因为考虑到项目类型是算法模型，所以在项目设计过程中，更多的投入于算法设计当中，对于用户体验并没有过多考虑，在设计数据输入模块时，更多的考虑了数据输入的便捷性，而没有考虑对于非法输入的限制，导致了当出现非法输入或输入格式不正确时，程序可能出现问题而停止运行，

规避方案：对于使用方式的说明书进行认真阅读，对数据的输入格式进行学习与熟悉，从而避免出现非法输入，减少出现问题的可能。后期对算法进行移植时，提升对用户体验的考虑，增加输入限制与用户提示，增加程序鲁棒性，从根本上解决问题。

6.2.4 在极端情况下，有极小几率出现非最优解

因为遗传算法对于初代群体的基因库有一定的依赖性，初代的基因库是随机产生的。在极端情况下，随机产生的初代群体的基因库是缺乏多样性的，因为过度单一，导致在产生新群体时，受到一定的限制，无法出现最优个体，虽然在变异过程中会增加一定的多样性，但是群体中变异的概率是较小的，因此每代增加的多样性是较小的，在变异增加的多样性达到可以起作用的程度之前，可能问题的求解已经结束，导致出现非最优解的情况。

规避方案：因为该极端情况出现的情况极小，因此正常情况下可以忽略此问题的存在，如果出现了明显的非最优解，最简单的方法对问题进行重新求解，从而规避该风险。

6.3 技术使用风险

6.3.1 路线图存储风险

由于我们的路线图使用的存储方式是使用 txt 的文本形式，所以如果遇到异常情况，存储路线图因恶意操作而被打开进行了删除等操作，会导致所存储的路线图文件出现损坏而无法复原，因此出现不必要的损失。

规避方案：对存储文件进行访问权限的设定，让非管理员无法打开该存储文件进行相关修改，同时对所创建的路线图文件，进行定期的备份工作，以便在出现文件损失等问题时，能及时找回文件，使得所导致的损失减到最小。后期在算法移植的过程中可以考虑更换存储方式，使用专业的数据库等方式，从根本上将问题缩小。

6.4 易用性及使用门槛

本系统的设计主要是为方便算法实现和结果展示，所以使用的易用性其实相对较高，因为没有较多的流程与设置，只是实现了单一的优化路线的功能，对于用户来说只要对使用说明进行基本的了解，便可以很快上手进行使用，所以使用门槛是相对较低的。

6.5 产品环境依赖性

因为本系统主要用于演示算法，只需要一台基本的计算机的可运行，所以对于环境的依赖并不是很高，只需要基本的管理员进行操作，将需要进行路线优化的路线图进行创建即可。

6.6 技术要求

操作系统	Windows 10
界面 UI	QT
开发语言	C/C++
主要技术	图的相关算法，遗传算法
数据库	TXT

7. 其他可行性分析

7.1 经济可行性

我们团队采用 vs+qt 设计 CVRP 软件，该软件的设计及安装都只需要一台通用电脑，其中设计的软件和技术都是开源或免费的，同时因为软件采用的技术不涉及大数据，因此在性能上也不需要搭建服务器，故而在整个研发过程中都不会设计到任何经费，因此在经济上可行。

7.2 社会可行性

7.2.1 市场可行性

(1) 配送在市场中的重要性

配送作为一种新型的物流手段，伴随着生产的不断发展而日趋成熟。发展配送，无论对于物流系统的完善，企业的发展，还是整个经济效益的提高，都具有重要的作用。

A. 配送完善和优化了物流系统。第二次世界大战后，高水平的干线运输呼唤支线运输和小搬运配套，但支线运输和小搬运在适应性、灵活性、服务性上的欠缺，致使运力不合理、运输成本过高。配送的出现使干线运输、支线运输及小搬运统一，输送过程得以优化和完善。

B. 配送提高了末端物流的效益。配送通过大批量进货，集中发货，以及将多个小批量集中一起大批量发货，都能有效的节省运力，实现经济运输，降低成本，使末端的物流经济效益得到提高。

C. 配送通过集中库存使企业实现低库存或零库存。采取准时制配送方式之后，生产企业完全可以依靠配送中心的准时配送而不需保持自己的库存或保持少量安全库存而不必留有经常库存。

D. 配送简化的事物，方便客服。采用配送的方式，客户只需向一处订购，或一个进货单位联系就可以订购到以往需要去许多地方才能顶到的货物，因此大大的减轻了客户的工作量和负担，也节省了事务的开支。

(2) 路线配送优化的市场意义

配送合理化与否是配送决策系统的重要内容，配送线路的合理与否又是配送合理化的关键。选择合理的配送路线，对企业和社会都具有很重要的意义。

对企业来说，1. 优化配送路线，可以减少配送时间和配送里程，提高配送效率，增加车辆利用率，降低配送成本。2. 可以加快物流速度，能准时、快速地把货物送到客户的手中，提高客户满意度。3. 使配送作业安排合理化，提高企业作业效率，有利于企业提高竞争力与效益。

7.2.2 社会因素的可行性

对社会来说，配送路线的规划和优化可以节省运输车辆，减少车辆空载率，降低了社会物流成本，对其他企业尤其是生产企业具有重要意义。与此同时，还能缓解交通紧张状况，减少噪声、尾气排放等运输污染，对民生和环境也有不容忽视的作用

7.3 实用可行性

随着互联网时代的推进，人们之间的现实距离越来越短，出行也越来越方便，但同时也让生活节奏变得更加紧凑，生活得碎片化时间也越来越多。这就促使了许多配送行业得出现，包括外卖，快递等等。

而经过我们团队得调研发现，现在越来越多得企业想要往配送行业方面发展，这就使得各企业对路线配送得规划和优化有了极大得需求。而我们团队研发得 CVRP 软件不仅所占内存小，同时得到得路线也能够满足绝大多数企业得要求，所以综上我们得项目能够在大多数企业运转，因此该项目使用可行。

7.4 法律可行性

本系统完全遵守国家及国际发布的法律法规，遵照中国颁布的《中华人民共和国著作权》与《计算机软件保护条例》中的内容，绝无侵权行为，坚决支持正版。使用 vs+qt 设计系统，均使用正版的操作系统软件进行开发，避免为此发生法律纠纷，因此本系统可以放心使用。

7.5 人员可行性

指导老师：指导老师是计算机专业，再算法方面有着丰富的工作经验，具备较强的教学和科研能力，曾主持或参与多项课题，具备较强的项目组织和指导经验，能对本项目实施的各环节给予全程指导。

本组技术成员都是计算机学院学生，都有一定的团队开发经验和实际项目的开发经验，并且有较强的自学能力和团队协作、团队沟通能力，对架构优化和系统性能有一定基础。

本项目由 HADEAN 团队，项目建设期历史 2 个月，虽然工作量较大，性能和安全性有很多优化的空间，但在团队人员得艰苦奋斗下，我们有信心也有能力在比赛限定时间内完成项目。因此本系统在人员上可行。