

**毕业设计（论文）**

基于区域聚类的多无人机与物流车路径优化算法研究

专 业 计算机科学与技术

学　　 号 1143730210

学 生 周昊天

指 导 教 师 曲明成

答 辩 日 期 2018/6/15

**摘 要**

车辆路径问题自1959年提出以来，由于其应用的广泛性和[经济](http://wiki.mbalib.com/wiki/%E7%BB%8F%E6%B5%8E" \o "经济)上的重大价值，一直受到国内外学者的广泛关注。带容量的车辆路径问题（CVRP）作为传统车辆路径问题中的一个重要分支，经过几十年的研究已经有了较为成熟的启发式算法优化方案。随着无人机技术的逐渐成熟以及在物流运输中的出众表现，带无人机的路径优化问题（UVAVRP）成为当前学术和工程界亟待解决的崭新问题。本文在考虑无人机与物流车协同配送这一现实场景下，对带无人机的CVRP进行了创新性的研究，结合使用改进后的聚类算法和禁忌搜索算法，对复杂问题进行预处理、全局CVRP求解、局部CVRP求解这三个阶段的分解，最终构建出用于解决该场景问题的三阶段混合算法，通过合理规划物流车与无人机的协同配送，有效地降低了运行成本和时间成本，从理论层面上创新性地解决了物流车与无人机协同配送这一难题。

关键词：带容量的车辆路径问题 ； 无人机 ； 协同运输 ；三阶段混合算法 ；聚类算法 ； 禁忌搜索算法 ； 局部分治

**Abstract**

Since its introduction in 1959, the vehicle routing problem has attracted widespread attention from scholars at home and abroad due to its extensive application and significant economic value. The capacity-carrying vehicle routing problem (CVRP) is an important branch of the traditional vehicle routing problem. After decades of research, there have been more mature heuristic algorithm optimization solutions. With the gradual maturity of UAV technology and outstanding performance in logistics and transportation, the path optimization problem (UVAVRP) with drones has become a new problem to be solved in the current academic and engineering world. In this paper, considering the cooperative scenario of unmanned aerial vehicle and logistics vehicle coordinated delivery, an innovative research is conducted on CVRP with drones, and the improved clustering algorithm and tabu search algorithm are combined to preprocess complex problems. , Global CVRP solution, local CVRP solution to these three stages of decomposition, the final construction of a three-stage hybrid algorithm to solve the problem of the scene, through the rational planning of logistics vehicles and unmanned aerial vehicles cooperative distribution, effectively reducing the operating costs and Time cost, from the theoretical level, innovatively solved the problem of collaborative delivery of logistics vehicles and drones.

**Keywords:** CVRP , UAV , cooperative transportation , three-stage hybrid algorithm , clustering algorithm , tabu search algorithm , local governance

目录

[第1章 绪 论](#_Toc4074_WPSOffice_Level1) [1](#_Toc4074_WPSOffice_Level1)

[1.1 课题背景及研究的目的和意义](#_Toc15585_WPSOffice_Level2) [1](#_Toc15585_WPSOffice_Level2)

[1.2应用场景描述](#_Toc17194_WPSOffice_Level2) [2](#_Toc17194_WPSOffice_Level2)

[1.3路径规划问题及其相关理论的发展概况](#_Toc31493_WPSOffice_Level2) [3](#_Toc31493_WPSOffice_Level2)

[1.3.1路径规划算法的发展](#_Toc15585_WPSOffice_Level3) [4](#_Toc15585_WPSOffice_Level3)

[1.3.2聚类算法与路径规划算法结合](#_Toc17194_WPSOffice_Level3) [5](#_Toc17194_WPSOffice_Level3)

[1.4 本算法特色与创新点](#_Toc4604_WPSOffice_Level2) [6](#_Toc4604_WPSOffice_Level2)

[1.4.1 聚类思想在本场景下的适用性改进](#_Toc31493_WPSOffice_Level3) [6](#_Toc31493_WPSOffice_Level3)

[1.4.2 三阶段混合算法对复杂问题的分解与整合](#_Toc4604_WPSOffice_Level3) [6](#_Toc4604_WPSOffice_Level3)

[1.4.3 引入无人机后的物流车与无人机协同配送方式](#_Toc30063_WPSOffice_Level3) [7](#_Toc30063_WPSOffice_Level3)

[第2章 问题模型构建](#_Toc15585_WPSOffice_Level1) [8](#_Toc15585_WPSOffice_Level1)

[2.1 问题场景描述](#_Toc30063_WPSOffice_Level2) [8](#_Toc30063_WPSOffice_Level2)

[2.2 数学符号定义](#_Toc26176_WPSOffice_Level2) [8](#_Toc26176_WPSOffice_Level2)

[2.3 数学模型构建](#_Toc17856_WPSOffice_Level2) [9](#_Toc17856_WPSOffice_Level2)

[第3章 问题模型求解](#_Toc17194_WPSOffice_Level1) [11](#_Toc17194_WPSOffice_Level1)

[3.1 算法思想总述](#_Toc18203_WPSOffice_Level2) [11](#_Toc18203_WPSOffice_Level2)

[3.2 各模块功能详解](#_Toc616_WPSOffice_Level2) [12](#_Toc616_WPSOffice_Level2)

[3.2.1预处理模块](#_Toc26176_WPSOffice_Level3) [12](#_Toc26176_WPSOffice_Level3)

[3.2.2 CVRP全局解模块](#_Toc17856_WPSOffice_Level3) [14](#_Toc17856_WPSOffice_Level3)

[3.2.3 CVRP局部解模块](#_Toc18203_WPSOffice_Level3) [15](#_Toc18203_WPSOffice_Level3)

[3.2.4 显示模块](#_Toc616_WPSOffice_Level3) [17](#_Toc616_WPSOffice_Level3)

[3.2.5 代价计算模块](#_Toc30100_WPSOffice_Level3) [19](#_Toc30100_WPSOffice_Level3)

[3.3核心算法解析](#_Toc30100_WPSOffice_Level2) [20](#_Toc30100_WPSOffice_Level2)

[3.3.1带限制条件的的迭代求解K-means聚类算法](#_Toc17224_WPSOffice_Level3) [20](#_Toc17224_WPSOffice_Level3)

[3.3.2 Tabu-CVRP算法](#_Toc24885_WPSOffice_Level3) [23](#_Toc24885_WPSOffice_Level3)

[3.3.3 局部类内分治规划算法](#_Toc15426_WPSOffice_Level3) [31](#_Toc15426_WPSOffice_Level3)

[第4章 实验及结果分析](#_Toc31493_WPSOffice_Level1) [33](#_Toc31493_WPSOffice_Level1)

[4.1实验简介](#_Toc17224_WPSOffice_Level2) [33](#_Toc17224_WPSOffice_Level2)

[4.2实验过程](#_Toc24885_WPSOffice_Level2) [34](#_Toc24885_WPSOffice_Level2)

[4.2.1 模拟数据生成模块](#_Toc2681_WPSOffice_Level3) [34](#_Toc2681_WPSOffice_Level3)

[4.2.2 样本数量测试](#_Toc9660_WPSOffice_Level3) [36](#_Toc9660_WPSOffice_Level3)

[4.2.3 轻重件比例测试](#_Toc1485_WPSOffice_Level3) [36](#_Toc1485_WPSOffice_Level3)

[4.2.4 无人机数量测试](#_Toc23681_WPSOffice_Level3) [36](#_Toc23681_WPSOffice_Level3)

[4.3 实验结果及分析](#_Toc15426_WPSOffice_Level2) [36](#_Toc15426_WPSOffice_Level2)

[结 论](#_Toc4604_WPSOffice_Level1) [37](#_Toc4604_WPSOffice_Level1)

[参考文献](#_Toc30063_WPSOffice_Level1) [38](#_Toc30063_WPSOffice_Level1)

[……](#_Toc28370_WPSOffice_Level2) [38](#_Toc28370_WPSOffice_Level2)

[……](#_Toc9577_WPSOffice_Level2) [38](#_Toc9577_WPSOffice_Level2)

[……](#_Toc2681_WPSOffice_Level2) [38](#_Toc2681_WPSOffice_Level2)

[哈尔滨工业大学本科毕业设计（论文）原创性声明](#_Toc26176_WPSOffice_Level1) [39](#_Toc26176_WPSOffice_Level1)

[致 谢](#_Toc17856_WPSOffice_Level1) [40](#_Toc17856_WPSOffice_Level1)

**第1章 绪 论**

## 1.1 课题背景及研究的目的和意义

9月5日，2017中国快递“最后一公里”峰会在北京举行，峰会指出中国快递业从2010年的10亿件发展到现在的400亿件。在快递业务量飞速增长，快递物流在人们的日常生活中扮演着十分重要的角色。

而快递运输中的“最后一公里”相关问题尤其突出：“最后一公里物流”是物流配送的最后一个环节，是指客户寄出的包裹，被运输到配送点后，由物流企业通过一定的运输工具将货物从最近的分拣中心送到客户手中，实现门到门的服务。物流成本是一种重要的生产经营要素，为了提高“最后一公里”的配送效率，降低配送成本是至关重要的。早在2010年中国物流与采购联合会发布的数据显示，中国物流总费用占国内生产总值比重约18%左右，其中“最后一公里”占到整个物流配送成本30%以上。刘强东在接受美国财经新闻网站CNBC的独家采访时提到，京东就雇佣了七万名快递员，如果能够用机器人（无人机）来送包裹，最后一公里配送的成本将会很低。

其次，最后一公里配送是城市配送环节中与客户直接沟通和接触的环节，这个环节的质量和效率很大程度上决定了客户的满意程度。配送员的众多不确定的个人因素很有可能成为影响客户体验的重要因素。使用无人车与无人机进行准时精确的配送，将毫无疑问提升用户满意度。

本课题组通过研究无人机控制与协同作业管理系统，要实现物流车与无人机协同作业的最后一公里配送自动化系统。该系统控制区域的分拣中心自动分配货物，发出携带无人机的物流车，实现物流车、无人机的协同作业完成区域快件的配送。该系统的完成将极大降低配送成本，提升区域快递最后一公里配送的效率并且提升用户满意度。

我在课题组中负责研究单派送中心多物流车多无人机情况下的路径规划算法的设计，该算法需要在综合考虑快件重量限制、无人车无人机协同工作等复杂条件下，规划出让综合行驶代价和总时间代价尽可能低的快件配送路径。

## 1.2应用场景描述

快件点（轻件点、重件点）：初始生成的快件点只有x坐标，y坐标和重量三个属性，根据无人机的载重进行划分，低于重量阈值的视为轻件点，大于重量阈值的视为重件点，轻件点可以由无人机或者物流车送达，而重件点只能由物流车送达。

模拟真实物流配送场景： 真实的物流配送过程中，快件点的产生总是成局部集聚的趋势（比如在城市中住宅区和商业中心快件点分布密集而公园、广场以及告诉公路沿路快件点稀疏甚至没有），以这个场景为前提，本文实现了模拟真实环境快件点分布的高斯函数模型模拟出了在一片区域中出现数个快件点呈正态分布的集聚中心，这些中心的边缘区域零散分布着少许的快件点。

单快件分发中心： 一片待处理的快件配送区域内只有一个快件分发中心，该中心存有多辆物流车可以并行执行配送任务。

物流车：每辆物流车带有容量限制，每次进行配送任务时能携带的快件重量之和不能超过物流车容量限制。同时每辆车配备多台无人机可以帮助物流车进行协同配送。

无人机：每架无人机也有着容量限制，每次飞行可以携带的轻件重件之和不能超过无人机容量限制。无人机有着作业半径限制，进行轻件点划分的时候将考虑到无人机的作业半径大小。

随着无人机技术的日益发展，众多国内外物流公司开始将无人机技术应用到物流配送行业中。亚马逊谷歌等公司首先将提出研发无人机空管系统的计划[9]。并于2016年利用无人机首次完成货物送达任务[10]。而在2017年首次提出利用地面车辆与空运无人机协同运输的货物运输方式[11]。然而随着无人机配送技术的日益成熟,物流车与无人机进行协同配送将成为可能，在传统的车辆路径问题重引入使用无人机还是一个崭新的领域。目前学术界对于物流车与无人机协同配送算法研究较少,现有的研究成果也都停留在的巡航等遍历问题上，如luo[14]等人采用启发式优化算法对区域内单车载无人机与地面车辆完成巡航任务的路线问题进行了优化方法，Yu[15]等人完成了单无人机与移动充电车辆配合完成区域内点遍历任务的广义旅行商问题（Generalized Traveling Salesperson Problem, GTSP）问题模型向TSP(Traveling Salesperson Problem，TSP）问题模型的转化。这些文献主要停留在完成对区域内点的访问任务，并没有将无人机作为运输载体，应用到物流配送这一场景中。

目前的传统物流配送算法都是只单纯基于车辆的路径规划，物流车有着载重量大的优点，然而配送速度慢、配送成本高并且受道路交通限制很多区域无法送达或者配送代价很大。而无人机虽然容量小，但是速度快运输成本低，不受地形与交通情况限制。如果结合两者的优点，考虑物流车与无人机的协同配送情况，就产生了全新的问题领域：带无人机的路径规划问题（UAV-VRP），本文从这一全新领域入手，创新性地提出了一种基于区域聚类的三阶段物流车与无人机协同配送优化算法，经大量实验证明相比较传统物流配送算法，能够同时有效降低行驶成本与时间成本。

## 1.3路径规划问题及其相关理论的发展概况

物流配送是物流系统的关键功能之一，涉及从制造工厂或配送中心通过运输网络到消费者的产品流动。运筹学研究文献称之为车辆路径问题（VRP）。 VRP是一种通用名称，指的是一类组合优化问题。其中客户将由多辆车辆提供服务。车辆离开仓库，为客户提供网络服务，并在完成路线后返回仓库。每个客户都有一定的需求。这个问题在Dantzig和Ramser（1959）的文献中首次提出。  
VRP是NP难题（NP-hard） [2]，迄今为止，这类问题中没有一个找到有效算法。学术界目前倾向于接受[NP完全问题](https://baike.baidu.com/item/NP%E5%AE%8C%E5%85%A8%E9%97%AE%E9%A2%98" \t "https://baike.baidu.com/item/NP-hard/_blank)（NP-Complete或NPC）和NP难题（NP-Hard或NPH）不存在有效算法这一猜想，认为这类问题的大型实例不能用[精确算法](https://baike.baidu.com/item/%E7%B2%BE%E7%A1%AE%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/NP-hard/_blank)求解，必须寻求这类问题的有效的[近似算法](https://baike.baidu.com/item/%E8%BF%91%E4%BC%BC%E7%AE%97%E6%B3%95" \t "https://baike.baidu.com/item/NP-hard/_blank)。因此，由于商业求解者需要解决的真实世界VRP的大小和频率，他们倾向于使用启发式算法。   
VRP在工业界有许多显著的应用。因为运输通常是产品成本的重要组成部分（10％）[4] ，VRP优化算法的使用通常可以为大多公司节省5％的成本[3]。事实上，交通运输行业占10％欧盟国内生产总值的百分比。因此，VRP节省的成本，即使不到5％，也是非常重要的。[3]s

https://en.wikipedia.org/wiki/Vehicle\_routing\_problem

1.3.1路径规划算法的发展

在形式上，经典车辆路径问题（VRP）由有向图G(E，V)表示，其中V = {0,1，...，n}表示节点集合，E是弧。 仓库被为节点j = 0，并且客户端是节点j = 1,2，...，n，每个节点需求dj> 0。每个弧表示从节点i到节点j的路线。 每个弧Cij> 0的权重对应于从节点i到节点j的成本（时间或偶数距离）。 如果Cij = Cji，那么我们正面对对称的VRP，否则问题是不对称的。 从复杂性的角度来看，经典的VRP是NP-hard已知的，因为它推广了旅行商问题（TSP）和Bin包装问题（BPP），它们都是众所周知的NP难题（Garey＆Johnson，1979）。 对经典VRP的数学公式的回顾可以在Laporte（1992）的着作中找到。

车辆路径（VRP）问题的研究起步早，经过50多年的研究，成果丰富。车辆路径问题最早由[1]Dantzig and Ramser(1959)提出，之后大量影响因素在这个问题中被考虑：路径长度（[2]Cordeau, Gendreau, Hertz, Laporte, & Sormany, 2005 ）、距离时间限制（[3]Juan, Faulin, Adelanteado, Grasman, & Montoya Torres 2009）、收取与分发、时间窗问题（[4]Lopez-Castro & Montoya-Torres, 2011）等等。车辆路径问题又根据研究方向不同衍生出 The capacitated VsRP ([5]Baldacci, Toth, & Vigo, 2010; [6]Cordeau, Laporte, Savelsbergh, & Vigo, 2007)；Split Delivery vehicle routing problem(SDVRP) ([7]Archetti & Speranza, 2008)；The vehicle routing problem with multiple depots（[8]Jairo R. Montoya-Torres, Julián López Franco, Santiago Nieto Isaza, Heriberto Felizzola Jiméne,& Nilson Herazo-Padilla, 2015）等问题。

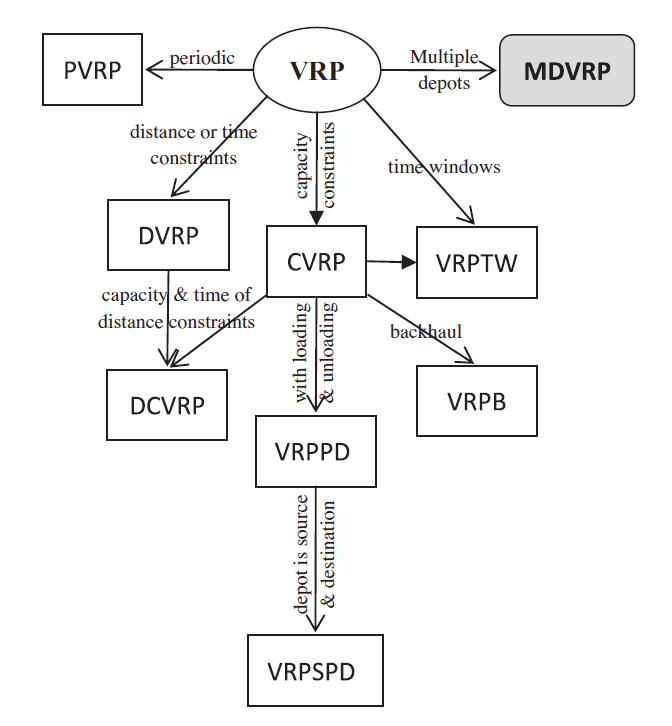
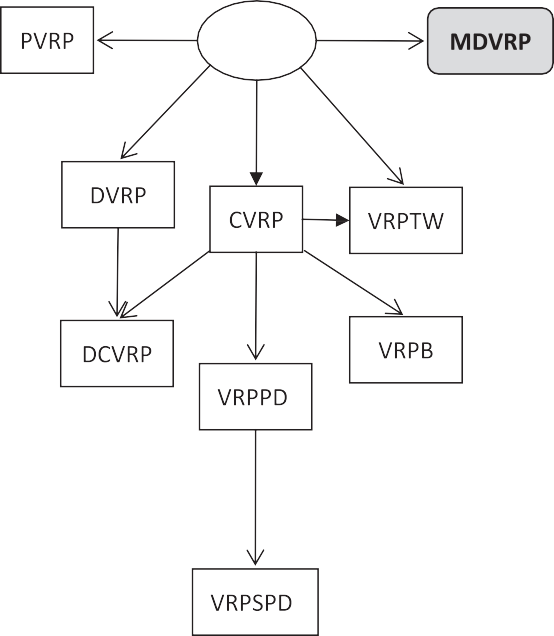


图 1-1 现有的VRP问题研究方向

虽然VRP问题又来已久，但是涉及两级车（two-echelon vehicle）的VRP问题却是一个崭新的领域，直到今年才刚开始出现相关的研究（[9] Xiaoling Zhong, Siyu Jiang, Hengjie Song, 2017 ;[10] Zhihao Luo , Zhong Liu and Jianmai Shi, 2017），然而这些文献的主要针对目标旨在于完成对区域内点的访问任务，并没有考虑无人机的承载能力和货物运输能力，物流车无人机进行物流协同配送是一块崭新的领域。



periodic

**VRP**

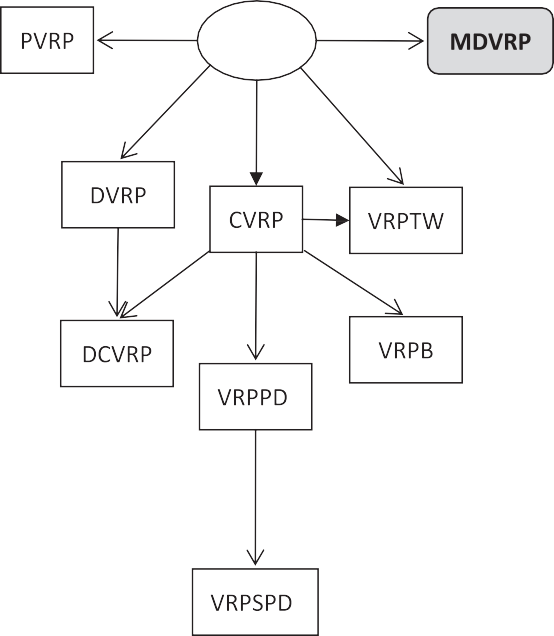
Multiple depots

distance or time constraints

time windows

capacity & time of distance constraints

backhaul



periodic

**VRP**

Multiple depots

distance or time constraints

time windows

capacity & time of distance constraints

backhaul

1.3.2聚类算法与路径规划算法结合

先聚类再规划路径可以通过将快件点预先分类再将每个聚类块视作整体进行路径规划。这种思想可以通过减少每次算法处理的点数来有效降低问题的复杂程度。目前已知的先聚类再规划路径的文献有[14,10,25]。

Gillett 1974 [14]是第一批采用先聚类再规划路径方法来解决VRP的作者。 他们开发了基于扫描的启发式算法，客户根据坐标以及容量大小确定不同的组别。每组数据当作TSP来解决。

Fisher 1981 [10] 提出了解决VRP的两阶段算法：第一个阶段中，根据特定标准筛选出每条路径中的快件点，接着第二阶段进行一个路径优化算法。

Qi 2012 [25] 发表的论文中综合考虑了空间和时间两个维度的信息对快件点进行聚类。他将代表时间和空间代价放到同一个坐标空间进行考虑，并且研究出一种测量两个快件点间时空间综合距离的方法。

## 1.4 本算法特色与创新点

1.4.1 聚类思想在本场景下的适用性改进

综合研究已有的先聚类后路径规划的算法思想之后，我发现由于都是应用在纯物流车配送的场景中，这些算法思想大多是划分完成运送区域后对区域内的所有点进行规划，再整体合并。这种先对小范围区域进行处理的方法，对聚类算法就有着较高的要求，聚类阶段就必须要在考虑到下一阶段整体规划的前提下进行合理划分，否则，先在局部完成该区域的路径规划很有可能会到导致整体性能的下降。

而在我们的应用场景中重件点与轻件点混杂，轻件点尽量使用无人机配送会大大降低配送成本。因此我们场景中的聚类对象为轻件点，聚类限制条件应为无人机覆盖半径。同时也要根据无人机数量对聚类区域的点数进行适当的初步限制。

而在聚类与路径规划的配合方面，与之前的大多数算法思想不同，本算法将采用先整体后局部的方式进行规划：首先对所有轻件点进行聚类，聚类完成后选取聚类中心，把原本的那个轻件点转换成这个类的等价重件点，附上整个类的快件重量。这样生成的等价重件点加入其余重件点中，视为重件点先用改进过的UGV-Tabu-CVRP算法进行一次CVRP求全局最优解，解出每辆物流车应该走的路线，然后再对每个区域内的轻件点进行UAV-Tabu-CVRP求解，解出局部无人机飞行路径的最优解。这样从整体细化到局部，首先保证了物流车代价的最小，其次，完美安排了无人机的最优配送路径，由于无人机配送速度快，区域内可与物流策划并行工作，时间代价主要就是物流车的配送时间。通过本文提出的先整体再局部的聚类-路径规划思想，理论层面上可以将综合代价降低至最小。

1.4.2 三阶段混合算法对复杂问题的分解与整合

本文待解决的应用场景涉及各物流车任务分配、无人机任务分配、物流车无人机容量限制以及物流车无人机协同工作等问题。算法的输入只有所有快件点的坐标和重量，要从简单的输入中提取信息，进行处理从而得到整体的路径规划，就需要对该复杂问题进行分解。我将整体算法主体分为主要三个阶段：数据预处理阶段、CVRP全局解阶段以及CVRP局部解阶段。

数据预处理阶段中，将输入的只有坐标和重量的原始数据先进行轻重件分离，再用改进的K-means聚类算法对所有轻件点进行聚类，并生成等价重件点，最后将在轻件点类内范围内的重件点归入该类。

CVRP全局解阶段中，将数据预处理阶段生成的等价重件点和其他重件点一起导入改进的UGV-Tabu-CVRP解决器中，输出每辆物流车分配的快件点路径。

CVRP局部解阶段中，将之前已经安排好的等价重件点还原成轻件点类，通过路径前后的类内重件点分布进行分类讨论处理先确定起飞点和着陆点，然后根据类内点数决定类内是进行单飞还是进行局部的UAV-Tabu-CVRP，最后输出每个类内的无人机规划好的路径。

1.4.3 引入无人机后的物流车与无人机协同配送方式

目前的无人机配送方式仅是定点起飞定点着陆进行配送，在无人机进行配送的过程中，物流车是停滞不动的，因此浪费了大量等待时间。但是考虑到安全因素以及技术条件限制，移动中无人机的起飞会合是不合理的。因此，受[XX]启发，设计出制定起飞点和着陆回合点的方式，这样能在物流车进行运输的时候尽可能并行派送无人机从而减少总代价。同时改进Tabu-CVRP算法，使其适用于非单点环路（双点单向路：起飞点和着陆点）的情况，实现每架无人机能在载重范围内携带多件快件的UAV-Tabu-CVRP算法。

**第2章 问题模型构建**

## 2.1 问题场景描述

本文的研究场景可以概括为：在仅有一个快件分发中心的区域中，有多辆（数量充足充足）容量为car\_w的物流车对m个快件收货点进行快递分发，且m个快件收货点上需要送达的总重量不同；其中总重量大于给定阈值K的快件收货点可划分为重件点，否则划分为轻件点，统称为快件点。

物流车携带分配给其的快件和n()架无人机从快件分发中心出发，尽可能以最小的行驶代价和最小的总时间成本，使物流车与无人机进行协同运输，将分配的所有轻重件点全部送达，最后回到快件分发中心。

无人机的配送范围要在其最大作业半径R以内，在单次随车配送过程中假设无人机电量充足，且能在物流车上实时充电。

## 2.2 数学符号定义

用到的概念转换为数学符号

|  |  |
| --- | --- |
| 符号 | 描述 |
| car\_w | 无人车最大载重 |
| car\_p | 无人车单位距离行驶成本 |
| car\_s | 无人车速度 |
| disij | 第i个快件点到第j个快件点的距离 |
| K | 无人机最大载重量（轻重件划分阈值） |
| class\_num | 类数量 |
| starti | 类i中的起飞点 |
| endi | 类i中的着陆点 |
| flight\_s | 无人机飞行速度 |
| fi | 无人机在类i中的路径数 |
| flight\_dij | 无人机在类i中第j条路径的飞行的距离 |
| flight\_wij | 无人机在类i中第j条路径上的快件点载重 |
| m | 快件点总数 |
| car\_ri | 给第i辆车分配的快件点集 |
| flight\_r |  |
| wi | 快件点i的快件重量 |
| yij | 物流车直接从快件点i运送到快件点j时置1 |
| cpi | 快件点i由物流车送达时置1 |
| opi | 快件点i由无人机送达时置1 |

## 2.3 数学模型构建

根据2.1中的问题场景描述，运用2.2中给出的符号定义建立本文要解决问题的数学模型：





 (1)

 (2)

 (3)

 (4)

Travel\_cost为物流车与无人机行驶代价之和即总行驶代价，通过将计算出的物流车与无人机路径长度总和分别乘以相应代价系数再求和就能得到总体的代价大小。

Time\_cost为物流车与无人机运行完所有快件所需的时间代价，计算方法为将物流车行驶需要的时间减去起飞点到着陆点间物流车行驶需要花费的时间再加上这两点间无人机最长路径需要花费的时间和物流车行驶时间的最大值。

本算法的优化目标就是在加入无人机的情况下尽可能降低这两者的大小。

约束条件（1）表明默认无人机配送速度比物流车快，且物流车配送代价比无人机高；

约束条件（2）表明每架无人机配送任务携带的快件总重必须不大于无人机最大载重量K；

约束条件（3）表明每辆物流车携带的快件总重量必须下雨不大于物流车最大载重量car\_w；

约束条件（4）表明每个快件点都必须送到且只由物流车或者无人机访问一次。

**第3章 问题模型求解**

## 3.1 算法思想总述

本算法解决了单快件中分发中心，多物流车多无人机进行协同工作完成快件配送任务规划的难题。本算法的核心是一个三阶段的混合算法，三个阶段分别独立设计出三个模块，三个不同模块通过生成的中间数据文件连结,前一个阶段的输出转为下一个阶段的输入，最终输出计算好的物流车路径和无人机路径文件。同时设计好的显示模块可以分别显示物流车路径图与无人机路径图，代价计算模块可以根据不同的要求计算出路径代价和时间代价。

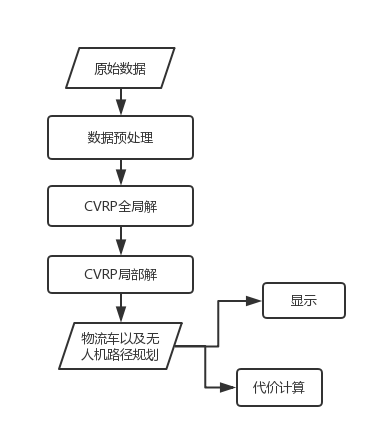


图 3-1-1 算法总流程图

## 3.2 各模块功能详解

接下来分别对各个模块进行详细地介绍

## 3.2.1预处理模块

预处理模块的功能总述：

先将输入的只有坐标和重量的快件点文件(Datasetx,xlsx)拆分成重件点（HP.xlsx）和轻件点文件(LP.xlse)并分别添加新的初始属性数据保存,然后运行计算路网矩阵的函数先生成轻件点路网矩阵；

接着进入此模块的核心部分将所有轻件点数据以及轻件点路网矩阵导入改进后的K-means聚类函数中，聚类函数对轻件点数据进行迭代处理直到每个类内数据都满足问题模型的限制要求，更新轻件点表的数据；

然后运行生成聚类中心的函数，在这个函数中，分别遍历每个类中的所有点选出到其他点距离的加权平均和最低的那个点设为该类的聚类中心即等价重件点，标记后添加到重件点表中，同时在遍历类内轻件点过程中判断上下左右边界，将类边界内的所有重件点都归为同一类并标记，更新重件点表；

然后生成现在的轻件-重件路网代价矩阵和重件点代价矩阵。

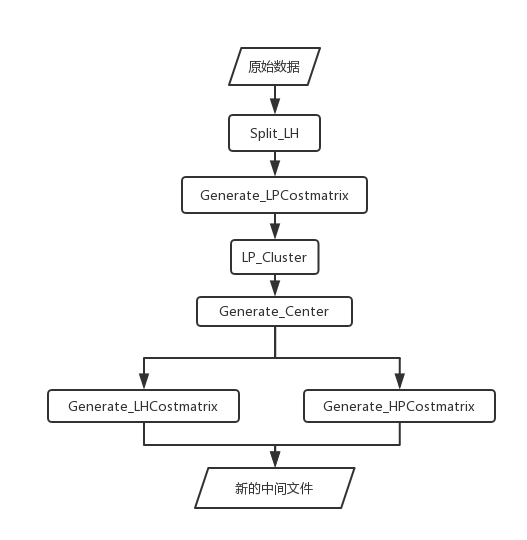


图3-2-1-1 数据预处理模块流程图

轻重件分离函数（Split\_LH）：

对于原始数据，轻重件分离函数将每个数据的重量属性与阈值K比较，不大于K的为轻件点，否则划分为重件点，分别添加type,class属性，并存入对应数据文件中。

轻件点路网代价生成函数（Generate\_LPCostmatrix，Generate\_LHCostmatrix，Generate\_HPCostmatrix）：

本文中的路网代价都是用欧式距离直接表示的，路网代价生成函数中， 对每两个点都进行遍历，计算出他们之间的欧式距离，并保存到路网代价矩阵中。

改进的K-means聚类函数（LP\_Cluster）：

这部分为预处理模块的核心算法，经过多种聚类算法的编写与测试，最后设计出了一种基于K-means聚类算法的带限制的迭代求解K-means聚类算法（将在3.3中详细解析），输入分离后的所有轻件点数据，聚类算法对轻件点进行迭代分类，每次迭代增加类的个数，直到所有类都满足聚类半径小于无人机工作半径且类内聚类点个数不超过给定值时退出迭代，根据分类结果更新轻件点表。

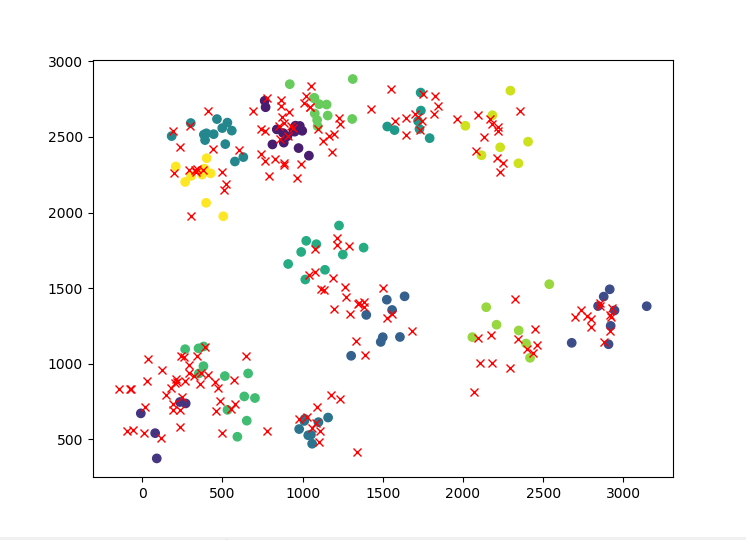


图 3-2-1-2 聚类模块效果图

生成聚类中心函数（Generate\_Center）：

首先，这个函数将划分好的每个类中所有点计算该点到其余所有点的加权平均和，取该值最小的点作为等价重件点会使该点到其余类内点的距离尽可能小且均匀。等价重件点确定后进行重新赋重（将整个类中轻件点的重量赋给该点）并添加相应的type，class属性并加入重件表中。

然后每类的边界也同时进行了判断，取类边界的方法也经过多次测试最终决定采用边界点左边进行矩形划分（使用以聚类中心为圆心的方案会导致大量点重叠，无法有效分类处理），类边界内包含的重件点将在类属性中相应地改变值，更新重件表。

最后就是导出初始化的Classtype表，这个表是为了CVRP局部解时记录每个类采用的无人机配送方式以及起飞着陆点而存在的。

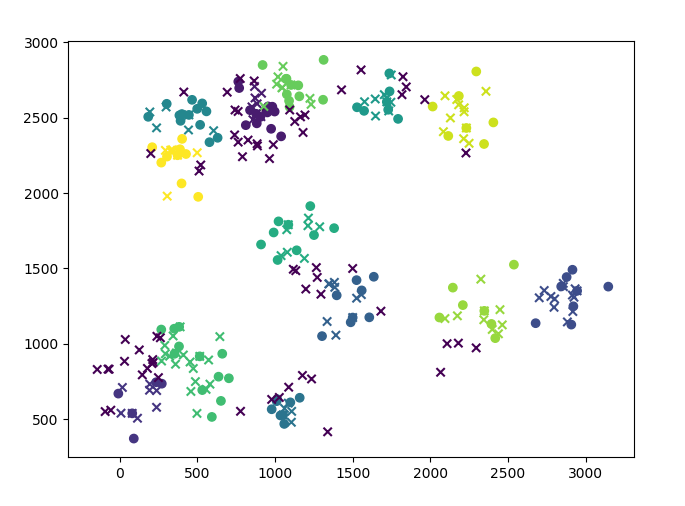


图 3-2-1-3 重件归类效果图

## 3.2.2 CVRP全局解模块

CVRP全局解模块用的是改进的禁忌搜索算法，禁忌搜索算法是启发式算法的一种也是本文的核心算法之一，将在3.3中对其进行详细介绍，本文先基本实现单程的禁忌搜索VRP解决器，再参考[]改进为CVRP问题解决器，由于禁忌搜索算法受初始解影响较大，再加上了多初始解并行运行取最优的机制，最终完成了UGV-Tabu-CVRP算法。该模块读取预处理模块处理好的带有等价重件点的重件表，对所有重件进行了CVRP规划，生成物流车的初步路径规划结果。

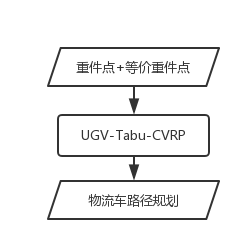


图 3-2-2-1 CVRP全局解模块流程图

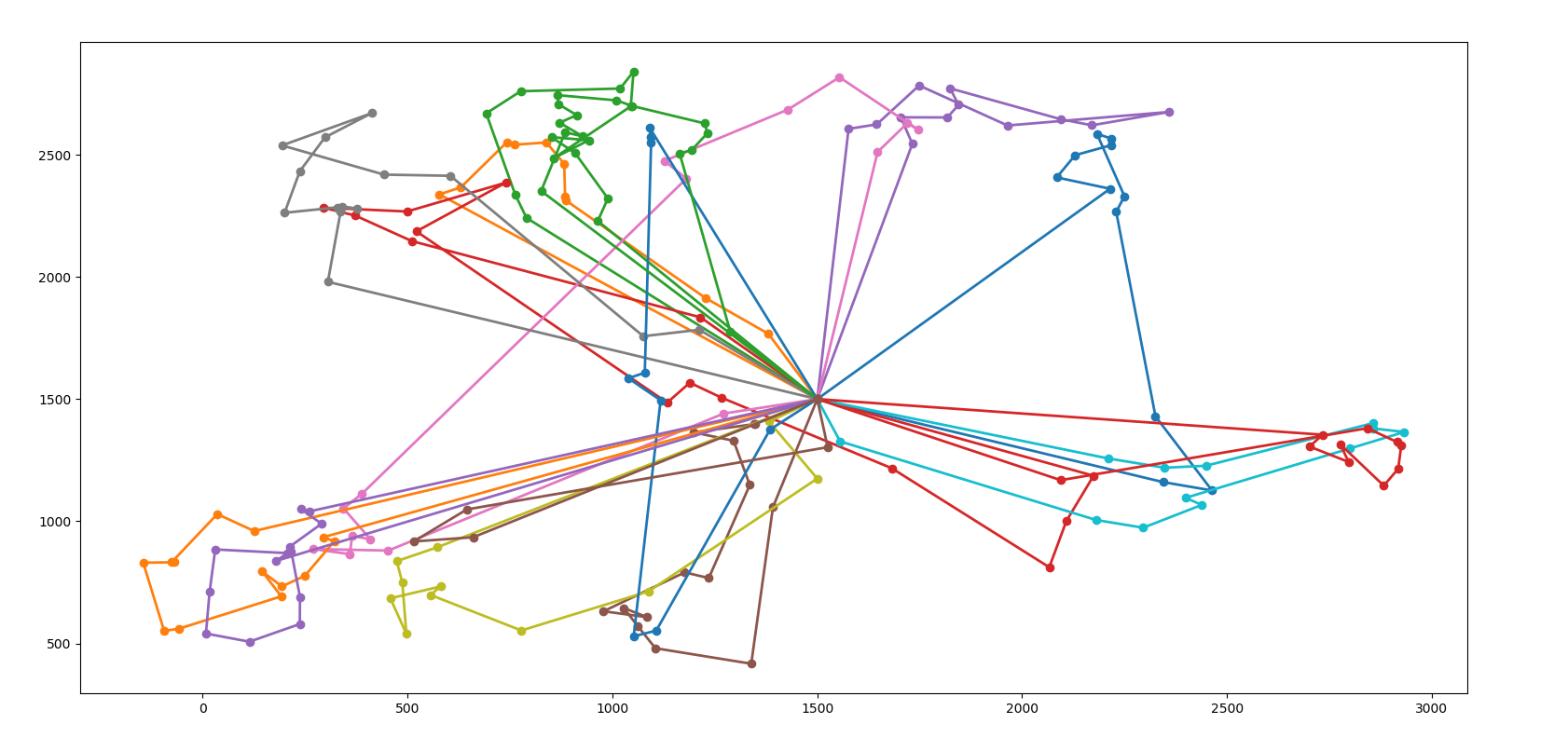


图 3-2-2-2 CVRP全局解模块规划效果图

## 3.2.3 CVRP局部解模块

CVRP局部解模块里由两个核心算法，一个是根据无人机派送情景改进的两点单向的UAV-Tabu-CVRP算法；另一个是对每个轻件点类内出现的所有的三大种、七小种情况进行了全面的分治，通过对所有情况的分类考虑，在类内点较多超过无人机数目时进行UAV-Tabu-CVRP求解尽可能降低无人机配送的行驶成本，在类内点较少时进行无人机直飞配送规划，使配送时间最短。（核心算法都将在3.3进行更细致的讨论）。

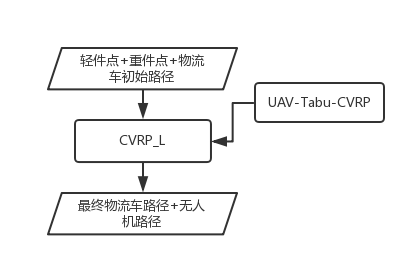


图 3-2-3-1 CVRP局部解流程图

UAV-Tabu-CVRP 算法：

由于应用场景和之前的全局CVRP情景不同（之前是单快件分发中心，规划出的每条结果都呈环路；而现在是从指定起飞点到终点的单向路径）数据结构与过程计算都要进行适用性大改。

CVRP\_L 分治算法：

由于轻重件点分布的随机性、初步路径规划结果的不可预测性以及尽可能减少总行驶代价的考量，根据每个类的具体情况对每个类的无人机规划策略进行划分。

从等价重件点出发向所在路径两边延伸，直到遇到的重件点不是该类类内重件点。

1. 两边都有类内重件点

两边的端点入口处设为起飞点，出口处设为着陆点

1. 轻件点数量小于无人机数量：不进行UAV-Tabu-CVRP局部求解每个点直飞配送
2. 轻件点数量大于无人机数量：进行UAV-Tabu-CVRP局部求解，输出路径结果
3. 两边都没有类内重件点
4. 类内轻件点小于等于无人机数量：单定点往返直飞配送
5. 类内轻件点大于无人机数量：

两两遍历所有类内点，找出两个轻件点使与路径前后两个最近的重件点路径长度之和最低（这样就能最大限度地降低物流车的行驶代价），转换为等价重件点，加入路径中，更新重件点表。

两边的入口处等价重件点设为起飞点，出口处等价重件点设为着陆点

1. 轻件点数量小于无人机数量：不进行UAV-Tabu-CVRP局部求解每个点直飞配送
2. 轻件点数量大于无人机数量：进行UAV-Tabu-CVRP局部求解，输出路径结果
3. 一边有类内重件点一边没有

有类内重件点的那头的重件点和等价重件点两者构成起飞和着陆点

1. 轻件点数量小于无人机数量：不进行UAV-Tabu-CVRP局部求解每个点直飞配送
2. 轻件点数量大于无人机数量：进行UAV-Tabu-CVRP局部求解，输出路径结果

## 3.2.4 显示模块

显示模块分为物流车路径显示模块和无人机路径显示模块。

两个模块都是先读取前面程序运行过程中生成的物流车和无人机路径结果文件New\_routes.csv或UAV\_routes.csv（通过python csv库里的reader函数逐行读取）

最后逐行存入数据矩阵中,再调用matplotlib里的绘图函数对每条路径进行绘制。效果如图。

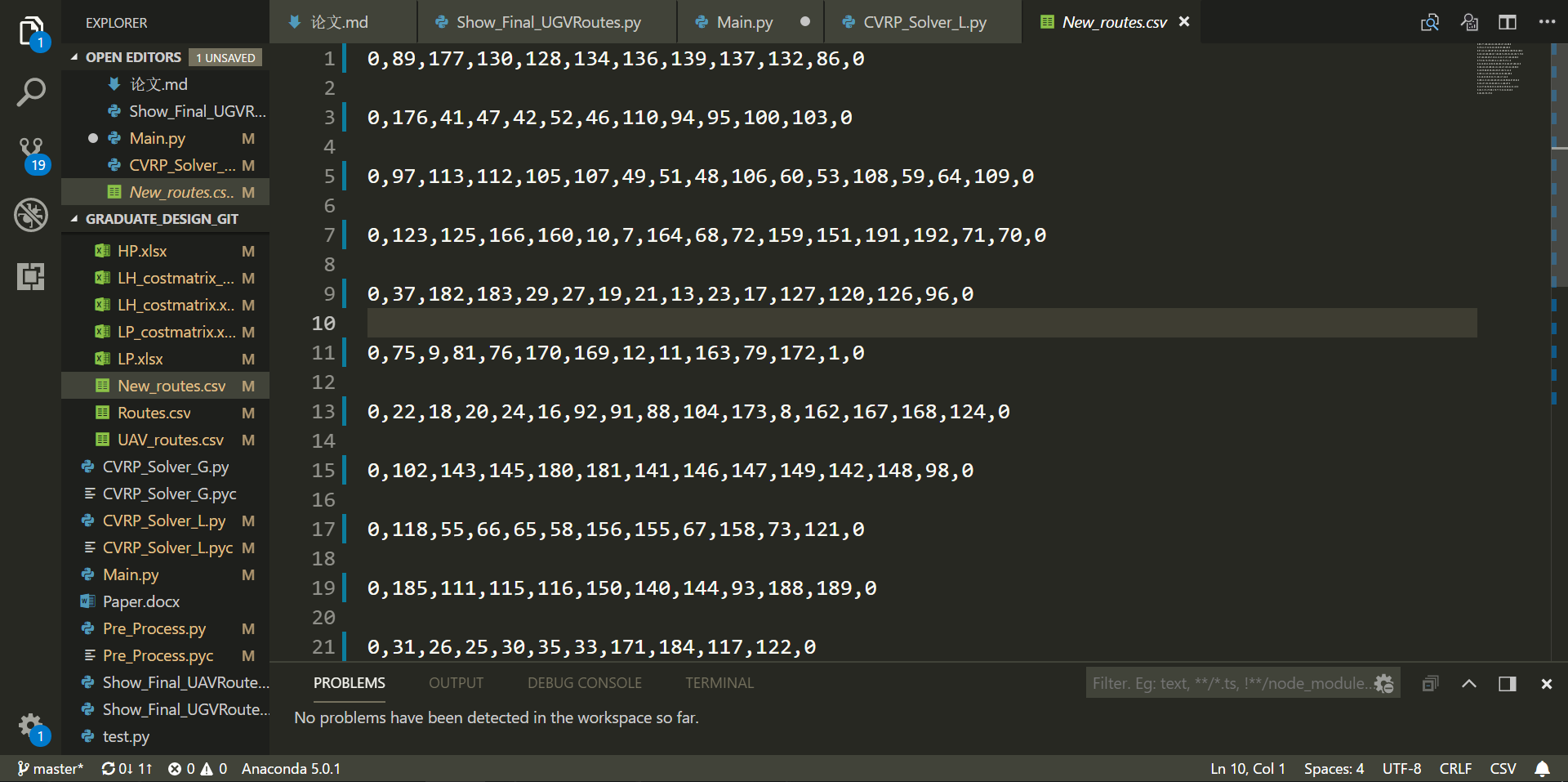


图 3-2-4-1 物流车路径文件

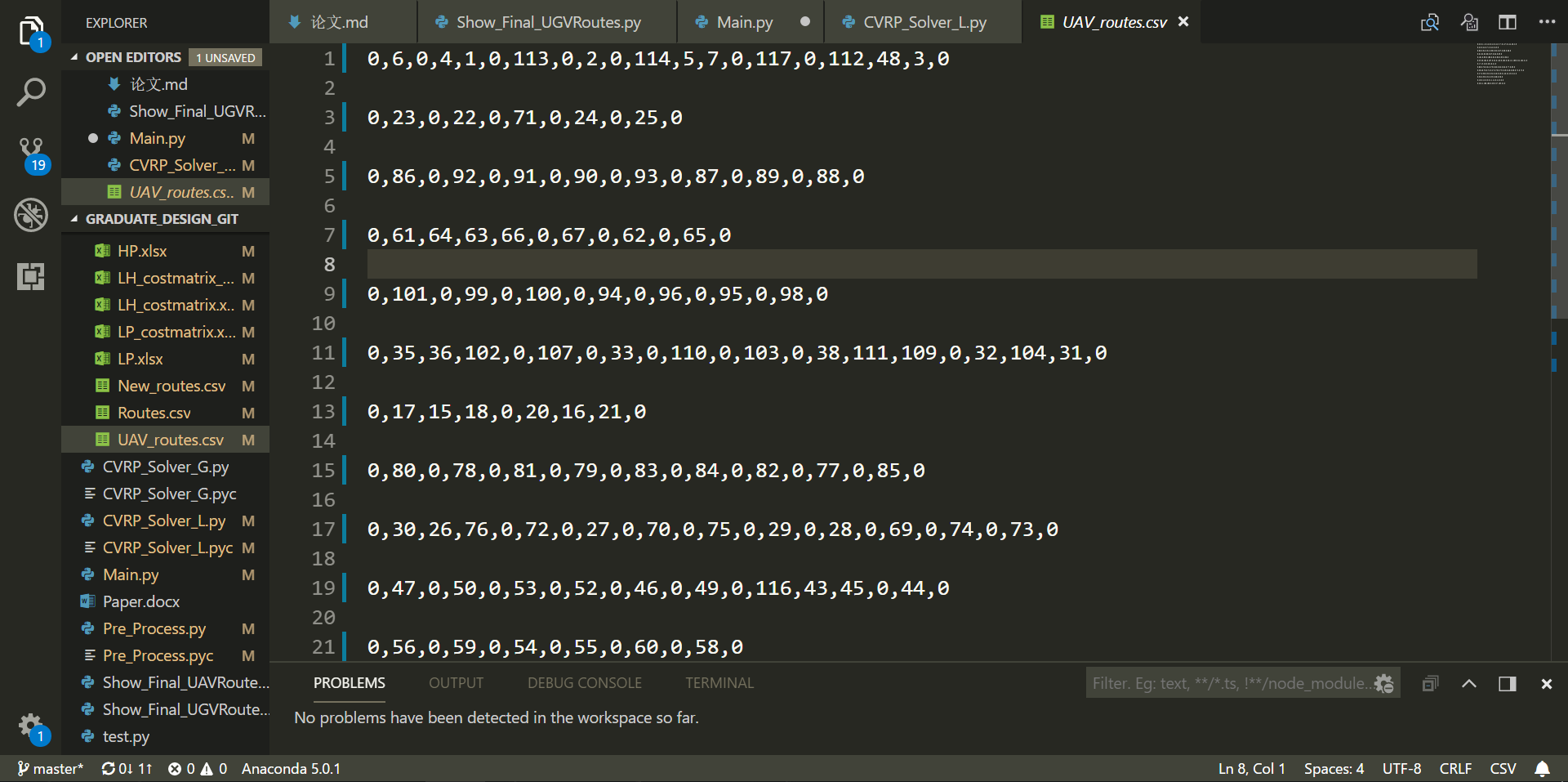


图 3-2-4-2 局部无人机路径文件

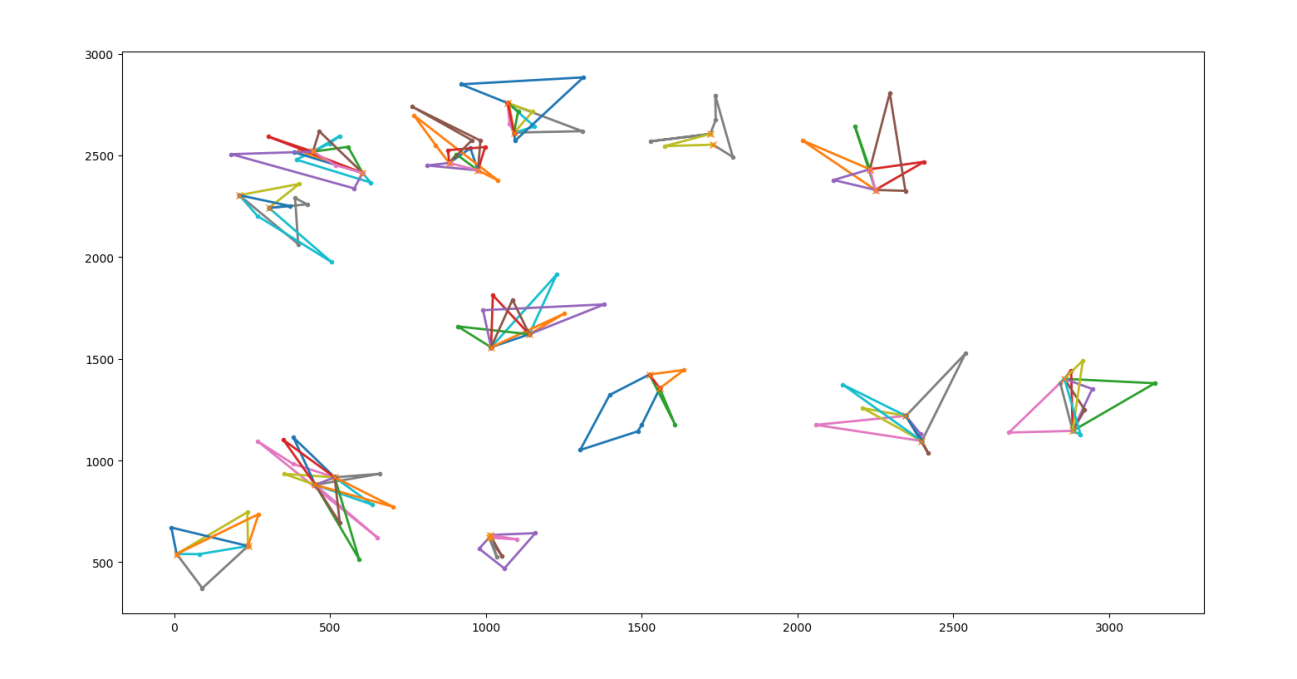


图 3-2-4-3 局部无人机路径规划结果图

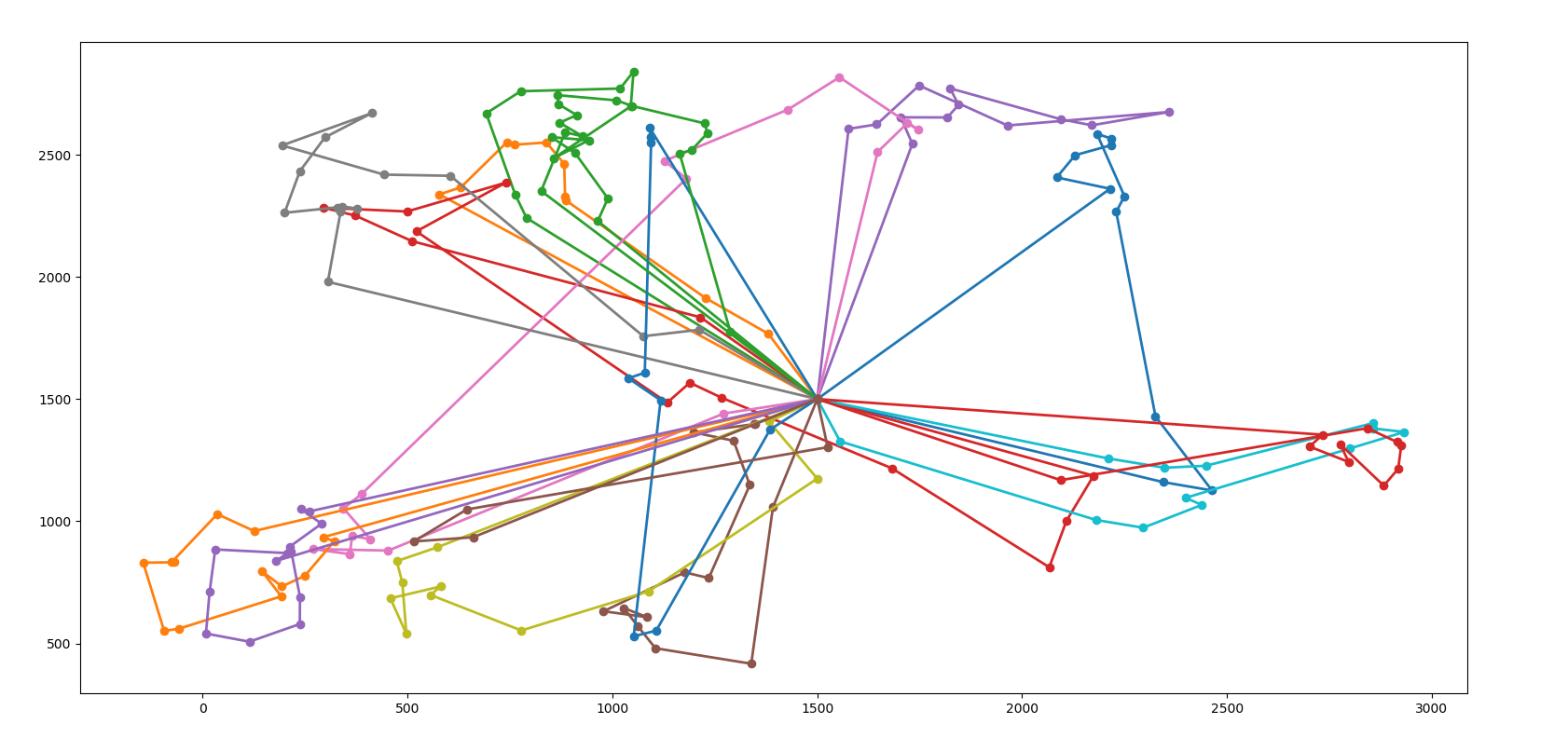


图 3-2-4-4 更新后的物流车路径图

## 3.2.5 代价计算模块

代价计算模块也是先读取前面程序运行过程中生成的物流车和无人机路径结果文件New\_routes.csv或UAV\_routes.csv（通过python csv库里的reader函数逐行读取），先计算所有路径长度之和，分别得到物流车行驶路径总长与无人机行驶路径总长，再根据问题模型求解里给出的公式，分别对两个计算到的路径之和乘以相应的代价系数，最终得到总的行驶代价。



时间代价计算比行驶代价计算要复杂，首先需要计算出物流车行驶的总时间（总路径长度除以物流车速度），然后减去起飞点和着陆点之间物流车行驶需要花费的时间，然后挑选出的每个类内无人机飞行的最远路径长度，计算出每个类内无人机路径最长的那条路径需要花费的时间（无人机路径长度除以无人机飞行速度）和物流车从每个类内起飞点着陆点花费时间的最大值。



## 3.3核心算法解析

## 3.3.1带限制条件的的迭代求解K-means聚类算法

本文中使用的聚类算法经过两次编写尝试。

## 初版分裂法聚类算法

初版的聚类算法，我在研究了各种聚类算法思想之后选择了分裂法聚类。分裂法聚类先将所有样本归为一个大的初始类，然后在不满足条件时反复分裂，直到满足限制条件要求时停止。由于我们场景中需要判断的限制条件是聚类半径大小和类内点的个数，这些条件可以在迭代分裂过程中轻易检测出来，这就使得聚类算法速度很快，但是却没有考虑我们聚类算法应用场景是局部点聚集的情况。分裂法聚类算法在样本分布均匀的情况下效果很好，但是在局部点聚集情况下却十分不理想。

分裂法聚类伪代码：

1.将样本集中的所有的样本归为一个初始类簇c0；

Repeat：

2.在同一个类簇（计为c）中计算两两样本之间的距离，找出距离最远的两个样本a, b；

3.将样本a，b分配到不同的类簇c1和c2中；

4.计算原类簇（c）中剩余的其他样本点和a，b的距离，若是dis(a)<dis(b)，则将样本点归到c1中，否则归到c2中；

Until：类内最远距离（最大飞行直径）

对初版的聚类算法进行测试，测试结果如图3-3-1所示：

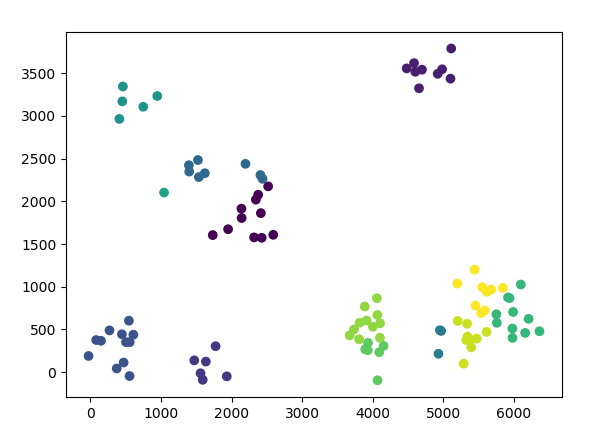


图3-3-1-1 分裂法聚类结果图

由于每次只是简单将点归于当前类的最远两个端点，首先只选取两个端点作为聚类标准就十分不科学，其次当出现两个最远点两有一侧点数量十分稀疏的时候，这个区域内的点，即使各自分隔稀疏，这个类的分裂也会结束，因此聚类的效果就很差，经过仔细思考后判断分裂法聚类不适用于本算法的场景。图中也可以看出即使点很少的情况下聚类结果还有明显的不合理点。

## 带限制的K-means迭代聚类算法

在经历了第一次聚类算法设计失败后，查阅了各种对于局部聚集情况聚类效果出色的聚类算法，最后决定了以K-means聚类算法为基础进行适用于本算法的改造。

K-means聚类伪代码：

1. 从数据中随机抽取k个点作为初始聚类中心

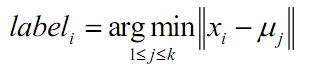
Repeat:

2、计算各个点到k个中心的距离，将其归到最近的聚类里

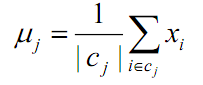
3、调整聚类中心，移到当前类的几何中心

Until:聚类中心不再移动

对于每个样本xi，将其标记为距离类别中心μj最近的类别，即：



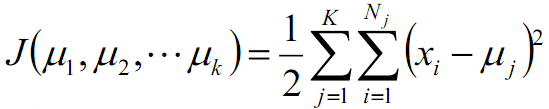
将每个类别中心更新为隶属该类别的所有样本(cj为j类中所有轻件点个数)的均值



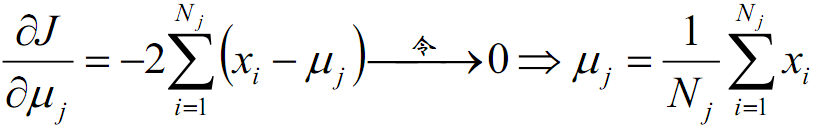
K-means算法公式化解释：

  记k个簇中心分别为μ1,μ2，μ3……μk,每个簇的样本数目为N1、N2……Nk。

使用平方误差做为误差函数，得：



将该函数做为目标函数，求解该函数的最小值。可以使用梯度下降法求，该函数为凸函数，驻点为：



可以看到，要想使损失函数最小，聚类中心要为各簇中样本点的平均值。由此可以看出，K-means算法在每次迭代更新时使用各簇中样本点的平均值为聚类中心是有道理的。

通过重复迭代K-means算法，逐渐增加类的个数直到满足限制条件（每个类的点数不大于预先设置的阈值以及聚类半径不大于无人机的工作半径。因为K-means聚类算法从聚类中心出发对所有点进行划分，并且在每次划分结束后进行聚类中心的重新选取，因此K-means算法在对集聚型数据进行处理时十分有效。K-means迭代算法实现效果如图（）

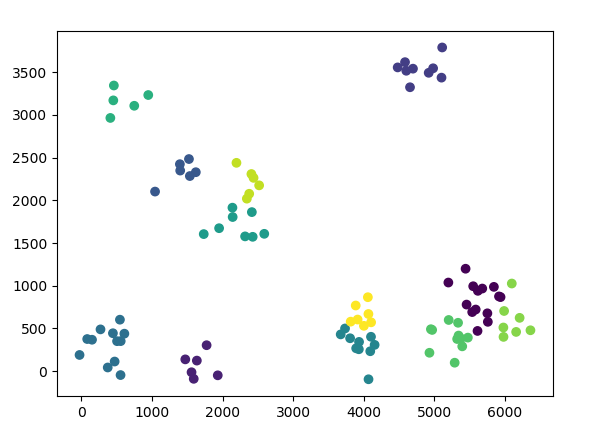


图3-3-1-2 限制条件为15点&1200聚类半径的K-means聚类迭代

图中可以看出每个类内的点都被很好的进行了划分并且都满足限制条件要求。

## 3.3.2 Tabu-CVRP算法

本文算法设计的另一个核心是以禁忌搜索（Tabu-Search）算法为基础的VRP解决算法。

在本算法设计过程中经历了

1. 用禁忌搜索算法实现单环路车辆路径规划算法（Tabu-VRP）
2. 用禁忌搜索算法实现带容量限制的车辆路径规划算法（Tabu-CVRP）
3. 改进禁忌搜索算法，最终得到适用于物流车配送的带容量限制的车辆路径规划算法（UGV-Tabu-CVRP）
4. 根据无人机场景中的起飞点着陆点不同，改造前面得到的Tabu-CVRP算法的数据结构得到适用于无人机配送的带容量限制的载具路径规划算法（UGV-Tabu-CVRP）

下面进行详细介绍：

1. Tabu-VRP

禁忌搜索算法伪代码：

1. 随机生成一个初始解X0设置为当前解Xnow，建立空的禁忌表

Repeat:

1. Xnow生成若干邻域解（2-交换、不能在禁忌表中出现）
2. 在Xnow领域中找出代价最小的解更新Xnow，更新禁忌表

Until:满足要求或者达到固定迭代次数

禁忌搜索（Tabu Search或Taboo Search，简称TS）的思想最早由Glover(1986)提出，它是对局部领域搜索的一种扩展，是一种全局逐步寻优算法，是对人类智力过程的一种模拟。其特点是采用禁忌技术，即用一个禁忌表记录下已经到达过的局部最优点，在下一次搜索中，利用禁忌表中的信息不再或有选择地搜索这些点，以此来跳出局部最优点。

在禁忌搜索算法中，首先按照随机方法产生一个初始解作为当前解，然后在当前解的邻域中搜索若干个解，取其中的最好解作为新的当前解。为了避免陷入局部最优解，这种优化方法允许一定的下山操作（使解的质量变差）。另外，为了避免对已搜索过的局部最优解的重复，禁忌搜索算法使用禁忌表记录已搜索的局部最优解的历史信息，这可在一定程度上使搜索过程避开局部极值点，从而开辟新的搜索区域。

禁忌搜索最重要的思想是标记对应已搜索的局部最优解的一些对象，并在进一步的迭代搜索中尽量避开这些对象（而不是绝对禁止循环），从而保证对不同的有效搜索途径的探索。禁忌搜索涉及到临域（neighborhood）、禁忌表（tabu list）、禁忌长度（tabu length）、候选解（candidate）、藐视准则（aspiration criterion）等概念

禁忌算法其主要特点是在搜索开始阶段，解的质量提高很快，随着搜索过程的继续，解的质量的提高速度逐渐放缓，甚至在很长的搜索阶段内解的质量没有太大提高，适合中小规模的NP问题求解，整体效率比较均衡。

小总结：

1. 基本思想——避免在搜索过程中陷入局部最优
2. 只进不退的原则,通过禁忌表实现
3. 不以局部最优作为停止准则
4. 邻域选优的规则模拟了人类的记忆功能

根据上述禁忌搜索算法的思想编写出本文实现的初版禁忌搜索算法代码，初版算法流程如下图：

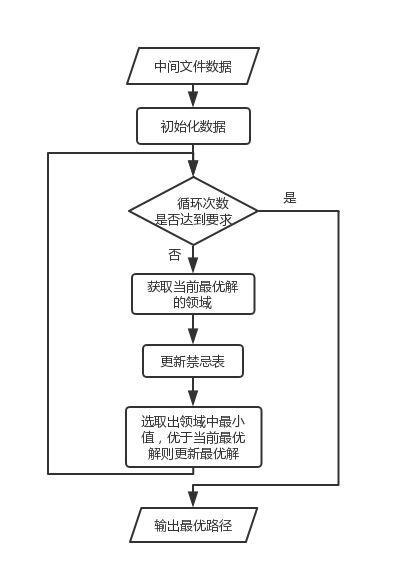


图 3-3-2-1 禁忌搜索算法流程图

中间文件的数据导入到禁忌搜索算法中后，算法先进行初始化，将所有算法要用到的全局变量赋初值；然后开始进入循环部分，每次循环中先生成当前的全局最优解（初始化中赋值为随机乱序的路径串）的领域解（采用随机2-opt交换方式）；然后取领域中的最优解与当前的全局最优解做比较，如果优于当前全局最优解的话，就用这个领域中的最优解更新全局最优解，然后继续开始下一次循环，直到循环次数满足要求。

禁忌搜索算法中的每一个解都代表着一种路径规划方案，循环结束得到算法计算出的全局最优解的时候，得到的解也就是单环路的路径规划结果。

实现基本的禁忌搜索算法后，运行TSPLIB att48测试集：

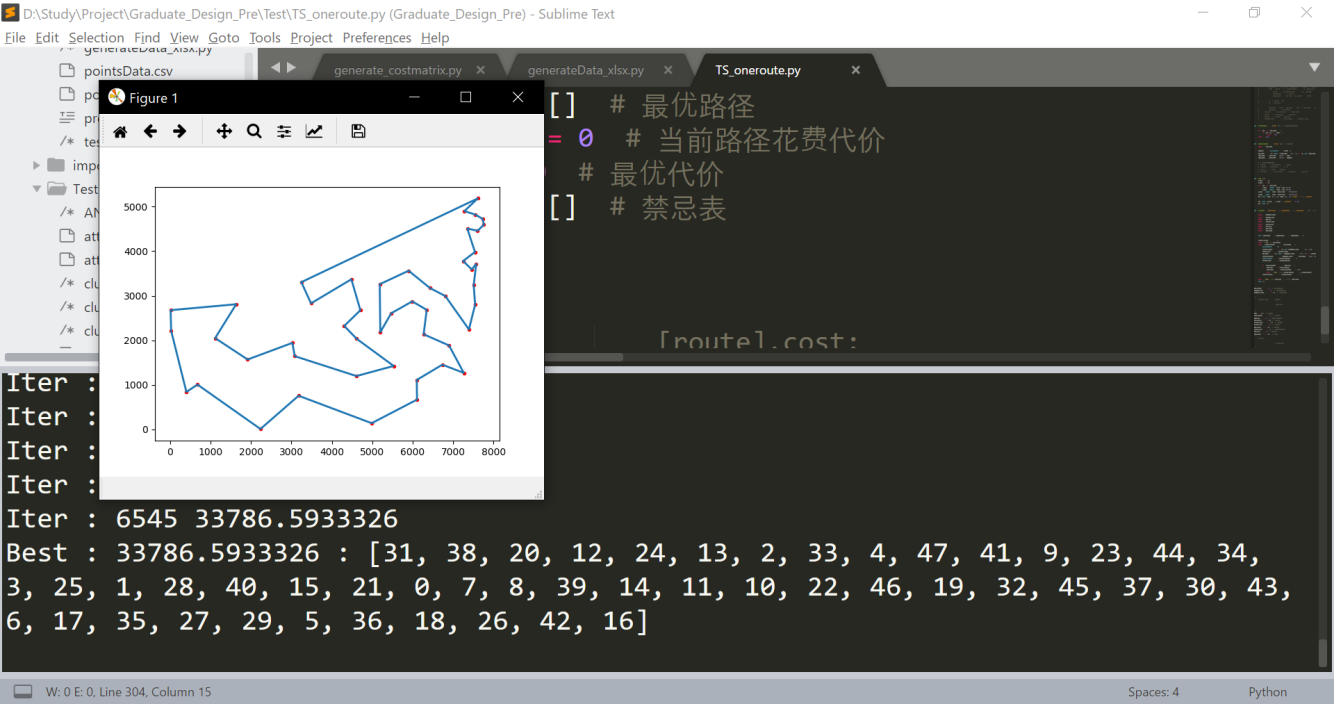


图3-3-2-2 单路程禁忌搜索算法 结果

Att48测试集的最优解是33522，实现的初版禁忌搜索算法最好能够达到33786的效果，已经很令人满意了。

1. Tabu-CVRP

如何将单路程的禁忌搜索算法改进为CVRP，是接下来需要考虑的问题。在单路程禁忌搜索算法中，每个解用一个不出现重复数的长数组List表示。

如何进行改造使Tabu-VRP算法也能应用于CVRP问题呢，这里参考了[]，我决定采用相同的策略，即通过在每次生成解的时候检索数字串，在当前扫描过的数字串所代表的快件点重量之和大于物流车容量的时候插入代表物流中心的0数字，扫描完成后该数列代表的真实路径规划就产生出来。然后再计算真实路径对应的代价。

这时编写算法改动的地方就是

1. 快件中心0加入重件点表
2. 编写生成真实路径函数，输入数字串后自动生成由0分隔开的真实路径
3. 生成领域函数中的生成代价函数改为计算由快件中心0分隔开的真实路径代价

改写Tabu-VRP算法，得到的新Tabu-CVRP算法运行结果如图所示

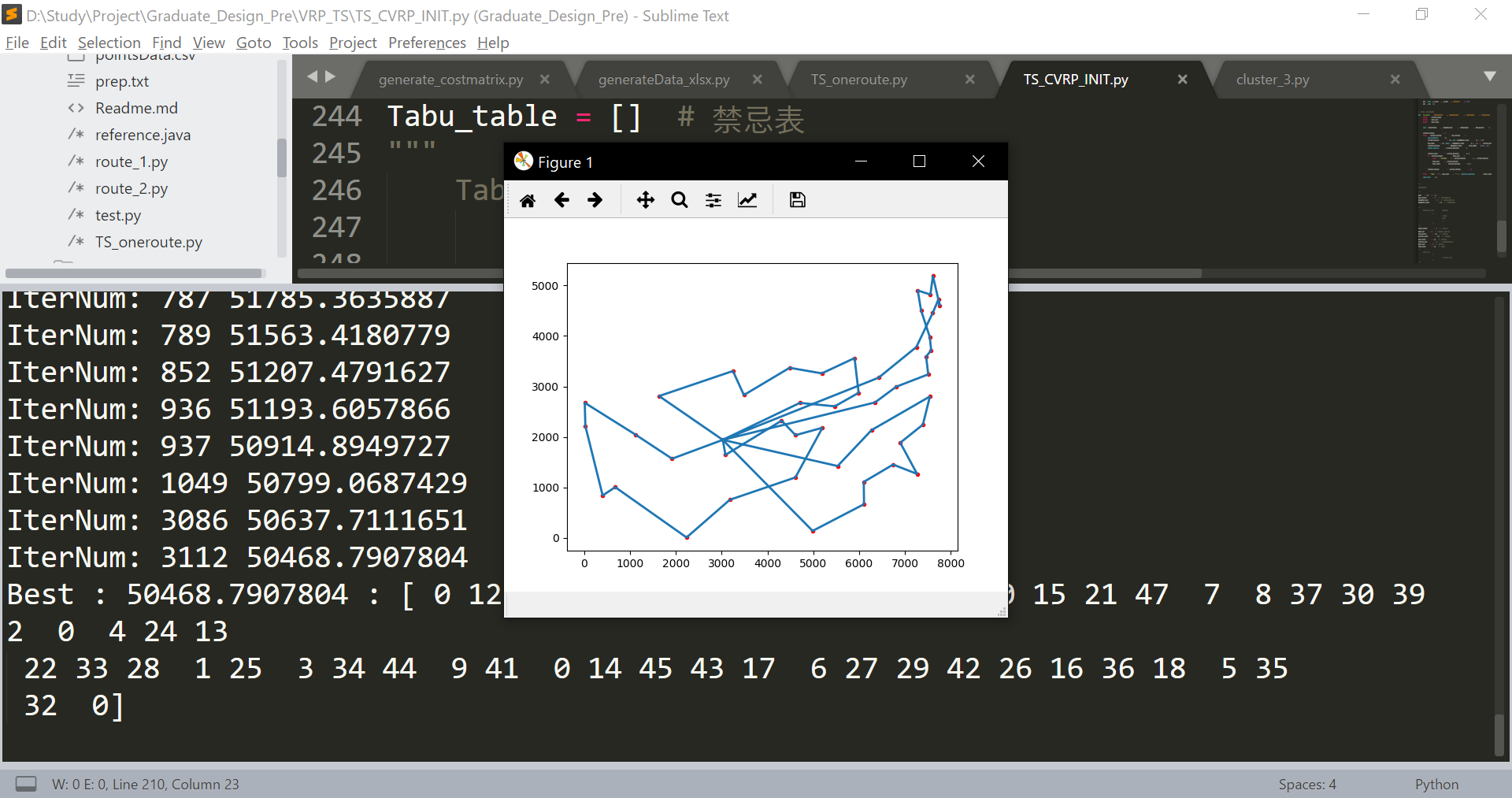


图3-3-2-3 Tabu-CVRP 结果图

规划后的路径已经出现了清晰的环路，总代价为50000以上，路径规划效果还行。但是算法很容易陷入局部最优，导致代价没有缩到最小，接下来就对这方面进行优化。

1. UGV-Tabu-CVRP

由于禁忌搜索是以初始解为根开始进行迭代的，算法得出的结果受初始解影响较大，不同的初始解可能导向不同的局部最优解。上面的Tabu-CVRP算法中采用的初始解就是一个随机生成路径序列转换成的真实路径，当算法开始运行的时候就注定了最终结果会过早地收敛到局部最优。因此，为了使算法结果代价尽可能的小，所以我决定使用多初始解，也就是随机生成多个初始解，同时进行迭代，迭代到一定次数后再选出当前最优秀的路径进行接下来的迭代，同时改进了路径显示函数，将每条路径分别导出再进行绘制，使结果更加清晰，同时加上路径上色，效果如图所示：

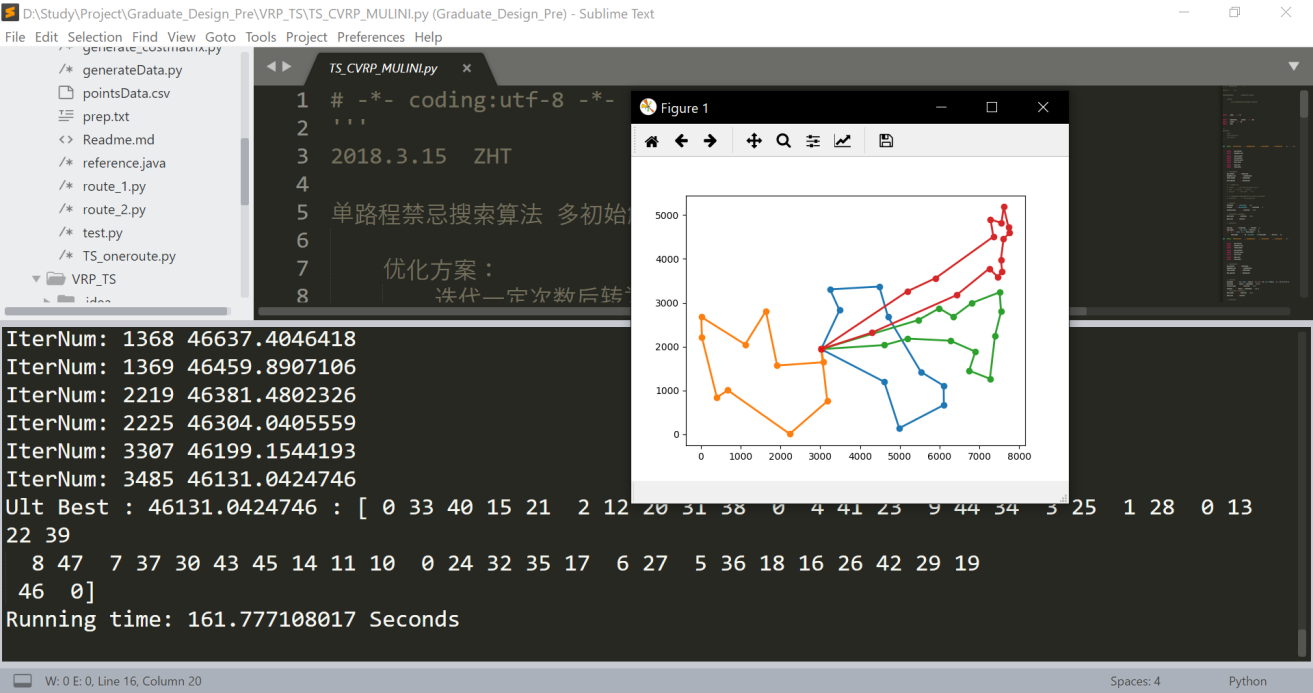


图3-3-2-4 UGV-Tabu-CVRP 效果图

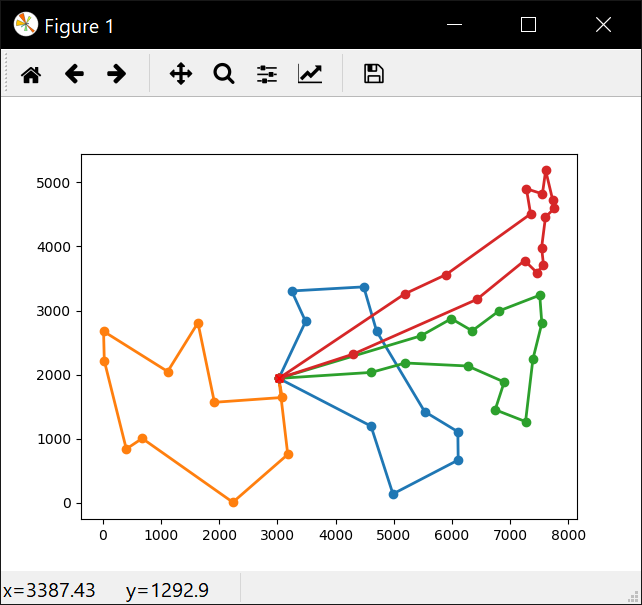


图3-3-2-5路径效果图放大

优化后的算法在运行同一测试集时代价降低到了46131，总体能比上个版本节约3000--4000的路径代价，规划出的路径也是十分清晰的环路，达到了预期效果。

1. UAV-Tabu-CVRP

先前考虑的Tabu-VRP与Tabu-CVRP都是基于车辆运输的情景。Tabu-CVRP情景中只有一个快件分发点，从该点出发的物流车完成配送后还要回到原来的位置。

而无人机模块的CVRP算法就大不相同，车载无人机是从起飞点统一放飞，各架无人机进行并行配送后飞往着陆点与物流车会合。大不相同的情景理应导致数据结构乃至算法的大改动，但是经过仔细地思考之后，我参考了先前Tabu-CVRP的解决方案，思考出了新的类似的无人机CVRP模块的解决方案:

同样在Tabu-VRP的基础上设置生成真实路径的函数，同样是当重量达限制条件后添加快件分发中心0，在将数字串生成真实路径之后代入新的计算代价函数中。

新的类内无人机路径代价计算函数:

1. 将每个0分隔出的中间数列分别取出，各自计算代价
2. 将每个中间数列都加上头尾轻件点分别到起飞点和着陆点的距离
3. 全部求和得到正确的当前路径规划解的总代价

其余算法部分与Tabu-VRP相同。

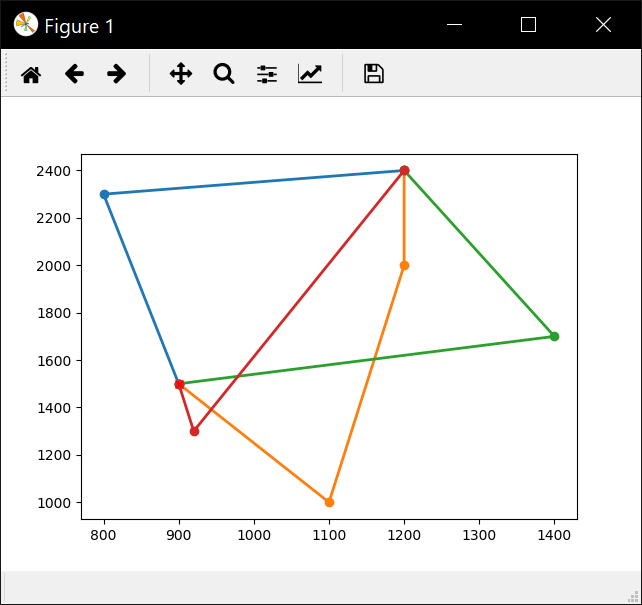


图3-3-2-6 无人机局部路径规划

## 3.3.3 局部类内分治规划算法

在算法的CVRP\_L模块中，进行由于轻重件点分布的随机性、初步路径规划结果的不可预测性以及尽可能减少总行驶代价的考量，根据每个类的具体情况对每个类的无人机规划策略进行划分。

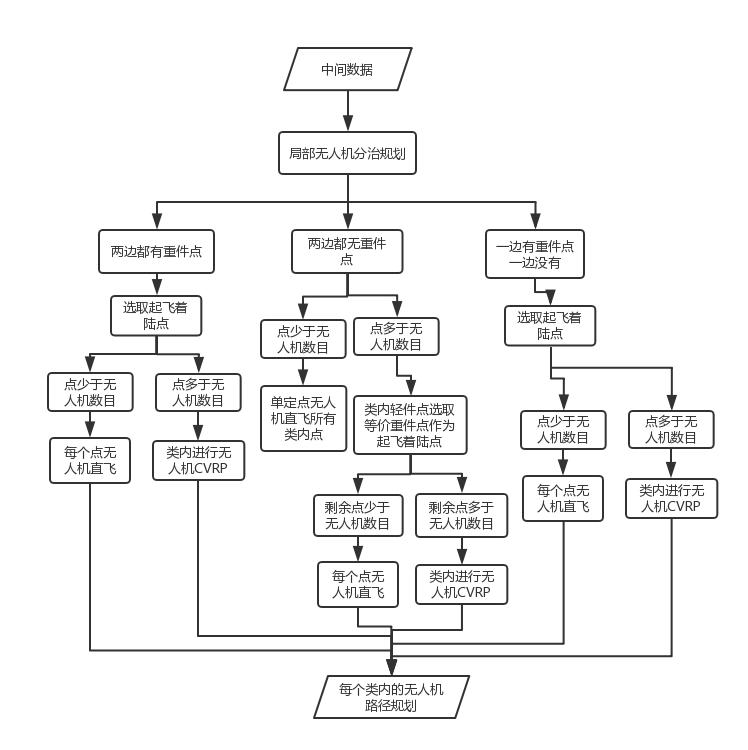


图 3-3-3-1 局部类内分治规划算法流程图

从等价重件点出发向所在路径两边延伸，直到遇到的重件点不是该类类内重件点。

1. 两边都有类内重件点

两边的端点入口处设为起飞点，出口处设为着陆点

* + - 1. 轻件点数量小于无人机数量：不进行UAV-Tabu-CVRP局部求解每个点直飞配送
      2. 轻件点数量大于无人机数量：进行UAV-Tabu-CVRP局部求解，输出路径结果

1. 两边都没有类内重件点
   1. 类内轻件点小于等于无人机数量：单定点往返直飞配送
   2. 类内轻件点大于无人机数量：

两两遍历所有类内点，找出两个轻件点使与路径前后两个最近的重件点路径长度之和最低（这样就能最大限度地降低物流车的行驶代价），转换为等价重件点，加入路径中，更新重件点表。

两边的入口处等价重件点设为起飞点，出口处等价重件点设为着陆点

* + - 1. 轻件点数量小于无人机数量：不进行UAV-Tabu-CVRP局部求解每个点直飞配送
      2. 轻件点数量大于无人机数量：进行UAV-Tabu-CVRP局部求解，输出路径结果

1. 一边有类内重件点一边没有

有类内重件点的那头的重件点和等价重件点两者构成起飞和着陆点

* + - 1. 轻件点数量小于无人机数量：不进行UAV-Tabu-CVRP局部求解每个点直飞配送
      2. 轻件点数量大于无人机数量：进行UAV-Tabu-CVRP局部求解，输出路径结果

经过这三大类7小类讨论，基本上所有情况已经都涵盖进去，运行完此算法后每个类内无人机的路径规划就都得以确定。

**第4章 实验及结果分析**

## 4.1实验简介

## 首先为了模拟本文算法所应用的局部集聚的真实情况，首先我设计出了模拟数据生成模块，该模块通过应用了高斯函数变化，可以自行设置大致的点聚类类数，集聚程度，快件重量范围，从而大致控制输出的局部集聚的模拟数据的分布情况。

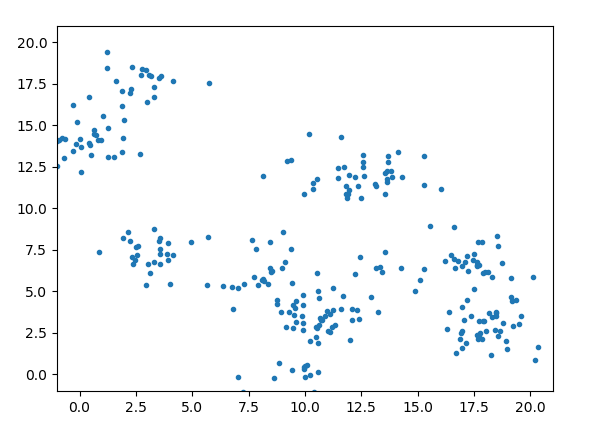


图 4-1-1 模拟数据生成模块效果图

测试阶段：

由于启发式算法受数据个数影响较大，作为程序的底层实现思想，有必要对不同数量的点数进行测试。设计的测试集分为50，150，250，300个点的生成的模拟数据。

其次是轻重件比例，本文研究的是物流车与无人机的协同 运输，轻件比例不同会很大程度上影响到包括聚类算法、局部无人机分治规划算法等一系列算法的性能，因此对其进行测试很有必要。

除了上面提到的两点，无人机的数目也对本文算法的运行结果有影响，当无人机数量增加时算法性能会得到提升。

## 4.2实验过程

实验平台介绍：

DELL XPS15 9550 CPU : Intel i7-6700HQ 2.6GHz

实验目的 :

1. 与传统物流算法对比，展示新算法优势
2. 算法自身测试，性能影响因素研究

实验过程设计：

1. 固定轻件占比40%左右，无人机数量4，分别生成50点左右，100点左右、200点左右以及300点左右集进行测试
2. 固定总件数为200件，调整轻件占比分别为20%，40% ，60% ，80%，固定无人机数量为4。
3. 固定总件数为200件，轻件占比40%，调整无人机数量为1，2，3，4，5测试代价随无人机数量上升的变化。

## 4.2.1 模拟数据生成模块

局部集聚的样本数据生成模块主要由两个函数构成：

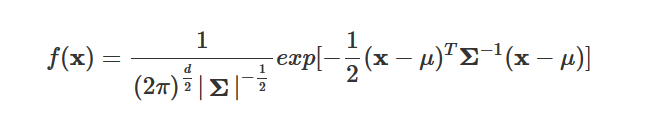
1. isDistinct

主要通过np.linalg.norm函数计算类间的二范数与给出的标准做比较，判断类与类之间的距离是否符合标准，是就返回True否则返回False。

1. generateData

生成数据模块中，首先根据给定的类数量随机生成多个类的中心点，同时用isDistinct函数判断避免靠的太近，然后根据设置好的高斯分布协方差在每个类中心点的周围生成样本数据。

多维高斯分布 ：



协方差矩阵的对角线元素为x和y轴的方差

反斜对角线上的两个值为协方差，表明x和y的线性相关程度（正值时：x增大，y也随之增大；负值时：x增大，y随之减小）

高斯分布的协方差矩阵是一个对称矩阵，该矩阵决定了二维高斯分布的形状，数值的大小决定了高斯分布的收敛程度。如图：

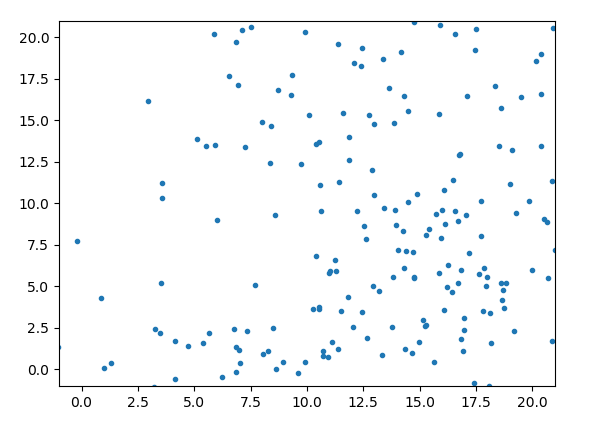


图 4-2-1-1高斯分布协方差[[10, 0], [0, 10]]

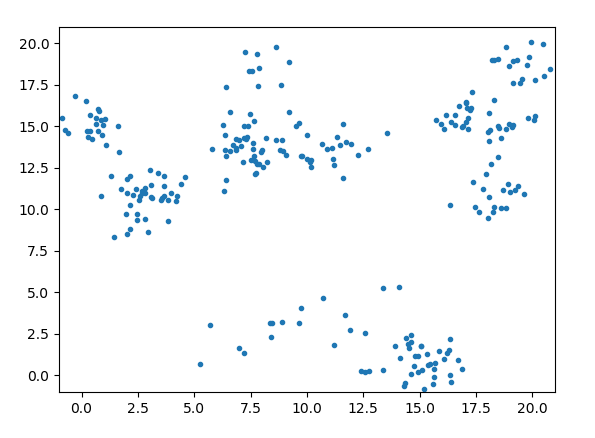


图 4-2-1-2高斯分布协方差[[1, 0], [0, 1]]

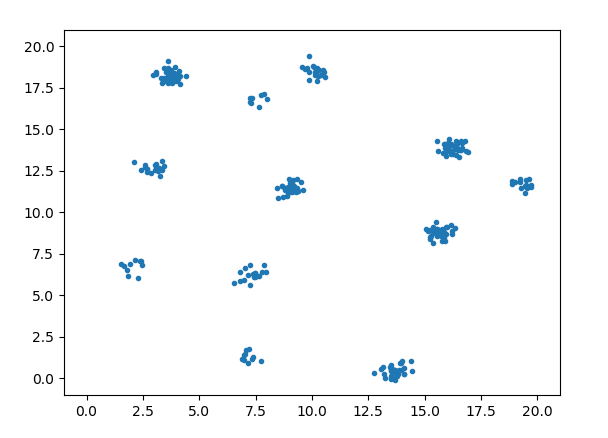


图 4-2-1-3高斯分布协方差[[0.1, 0], [0,0.1]]

## 4.2.2 样本数量测试

290点数据 Dataset4 结果：

## 4.2.3 轻重件比例测试

## 4.2.4 无人车载重测试

4.2.5 无人机速度测试

## 4.3 实验结果及分析

**结 论**

学位论文的结论作为论文正文的最后一章单独排写，但不加章标题序号。

结论是对整个论文主要成果的总结。在结论中应明确指出本研究内容的创新性成果或创新点（含新见解、新观点），并指出今后进一步在本研究方向进行研究工作的展望与设想，上述各项用（1）.（2）.表述，不要将结论写成论文的摘要。结论字数一般在2000字以内。

**参考文献**

[1] 林来兴. 空间控制技术[M]. 北京：中国宇航出版社，1992：25-42.

[2] 辛希孟. 信息技术与信息服务国际研讨会论文集：A集[C]. 北京：中国科学出版社，1999.

[3] 赵耀东. 新时代的工业工程师[M/OL]. 台北：天下文化出版社，1998 [1998-09-26]. http://www.ie.nthu.edu.tw/info/ie.newie.htm（Big5）.

……

[12] 谌颖. 空间交会控制理论与方法研究[D]. 哈尔滨：哈尔滨工业大学，1992：8-13.

[13] Kanamori H. Shaking Without Quaking[J]. Science，1998，279（5359）：2063-2064.

……

[104] Christine M. Plant Physiology: Plant Biology in the Genome Era[J/OL]. Science，1998，281：331-332[1998-09-23]. http://www.sciencemag.org/cgi/ collection/anatmorp.

……

**哈尔滨工业大学本科毕业设计（论文）原创性声明**

本人郑重声明：在哈尔滨工业大学攻读学士学位期间，所提交的毕业设计（论文）《 》，是本人在导师指导下独立进行研究工作所取得的成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明，其它未注明部分不包含他人已发表或撰写过的研究成果，不存在购买、由他人代写、剽窃和伪造数据等作假行为。

本人愿为此声明承担法律责任。

作者签名： 日期： 年 月 日

**致 谢**

衷心感谢导师×××教授对本人的精心指导。……，他的言传身教将使我终生受益。

感谢×××教授，以及实验室全体老师和同窗们的热情帮助和支持！

本课题承蒙××××基金资助，特此致谢。

…