

清 华 大 学

# 综 合 论 文 训 练

题目：考虑视觉合理性的异质车辆路  
径规划研究

系 别：工业工程

专 业：工业工程

姓 名：庄成

指导教师：王晓芳老师

2023 年 5 月 28 日

## 中文摘要

传统路径规划问题关注包括最短行驶里程、最短行驶时间、或车辆资源分配等客观的优化目标，能够生成效率高、时长短的路径规划方案。许多企业指出单纯考虑传统优化指标的路径规划的方案会出现线路交叉，线路重叠等不符合司机作业习惯的路径方案。为此，本文将视觉合理性这一主观指标加入到传统路径规划中，使路径规划方案融合司机作业偏好和人工路径规划经验。

本文以山西省某市的垃圾运收作业数据为例，通过构建考虑视觉合理性的数学模型，满足各项企业运营约束，利用启发式算法优化车辆作业路径的视觉合理性。

本文设计了一种以节约算法与引导式邻域搜索为基础的两阶段启发式算法，并通过算例对算法的可靠性和鲁棒性进行了验证。本文对比分析了传统路径规划与考虑视觉合理性的路径规划的各项评估指标。考虑视觉合理性和成本的双目标的路线方案，能够在保持低线路行驶成本的同时，更符合调度员和司机作业习惯。本文所采用的车辆路线规划方法为企业解决路径规划问题提供了参考。

**关键词：**路径规划；视觉合理性；启发式算法

## Abstract

Traditional Vehicle Routing Problem (VRP) focus mostly on more objective optimization goals, including minimizing route length, minimizing route time, or overall workload distribution. Although these traditional models are able to generate optimized routing plans, many companies have pointed out that these plans are not implantation friendly. These objective goals might results in route intersection, route overlap, or other instance that contradict the manual vehicle routing logic. Therefore, this paper has proposed a new subjective consideration, Visual Attractiveness. From a company's point of view, it is vital for vehicle routing algorithms to generate a more visual attractive routing plan while maintaining a minimum cost.

By utilizing real-time operation data from a garbage recycling firm from Shanxi province, including demand points, fleet information, and facilities, this paper has developed a mathematical model that took visual attractiveness into consideration while fulfilling other operation constraints.

Furthermore, referencing the savings method and guided local search algorithm, this paper has also proposed a two-phase heuristic algorithm aiming to optimize both the objective goals and visual attractiveness. Through calculating results on multiple dataset and analyzing the results, this paper has verified the reliability and robustness of the proposed algorithm. Moreover, by comparing results and key indicators between the visual attractive routing plan and the traditional routing plan, this paper has validated the necessity of including the visual attractiveness consideration in vehicle routing problems. The proposed method used in this paper is able to increase visual attractiveness of routing plans while maintaining a minimum cost, which can be used as future reference in company vehicle routing related implementation.

**Keywords:** Vehicle Routing Problem; Visual Attractiveness; Heuristic Algorithm

# 目录

第 1 章 引言 .....	1
1.1 研究背景 .....	1
1.2 研究意义 .....	2
1.3 研究内容 .....	3
1.4 论文结构 .....	5
第 2 章 国内外研究综述 .....	6
2.1 路径规划问题 .....	6
2.2 VRP 求解方法综述 .....	6
2.3 视觉合理性优化目标 .....	7
2.4 视觉合理性指标 .....	8
2.4.1 紧凑性 .....	8
2.4.2 非交叉 .....	9
2.4.3 低复杂度 .....	10
2.5 视觉合理性约束实现方法 .....	11
2.5.1 聚类法 .....	11
2.5.2 启发式插入算法 .....	14
2.5.3 邻域搜索 .....	15
2.5.4 引导式局部搜索 (Guided Local Search) .....	16
2.6 本章小结 .....	18
第 3 章 数学模型 .....	20
3.1 符号定义 .....	20
3.2 考虑视觉合理性数学模型 .....	21
第 4 章 算法设计 .....	24
4.1 两阶段算法整体介绍 .....	24
4.2 第一阶段——初始解算法 .....	25
4.3 第二阶段——视觉合理性优化算法 .....	27
4.3.1 线路紧凑性优化 .....	27
4.3.2 重叠摧毁重构器 .....	30

4.3.3 引导式局部搜索 .....	31
第 5 章 案例分析 .....	37
5.1 垃圾运收作业背景 .....	37
5.2 案例背景 .....	37
5.3 考虑视觉合理性车辆路径规划计算结果与分析 .....	39
5.3.1 C-W 节约算法计算初始解.....	39
5.3.2 紧凑性优化 .....	40
5.3.3 重叠摧毁重构器 .....	46
5.3.4 引导式局部搜索优化算法计算结果 .....	49
5.4 仅考虑行驶成本车辆路径规划问题计算结果与分析 .....	52
5.5 算法验证 .....	55
第 6 章 总结与展望 .....	58
6.1 研究工作总结 .....	58
6.2 研究结论 .....	58
6.3 未来研究展望 .....	59
插图索引 .....	60
表格索引 .....	62
文献参考 .....	63
致谢 .....	65
声 明 .....	错误!未定义书签。
附录 A: 外文资料的调研阅读报告 .....	67

# 第1章 引言

本章主要介绍了车辆路径规划问题的研究背景、研究意义，以及基于传统路径规划的创新考虑。同时，本章介绍了所使用的企业案例背景和问题，最后给出了本文的研究内容和架构。

## 1.1 研究背景

随着社会的发展，消费者生活节奏越来越快，对快递、外卖、出行、及其他配套生活服务的需求大大增加。据统计，中国在 2021 年外卖订单量超 200 亿单，单日外卖订单量可达 5500 万单；垃圾清运量达到 2.67 亿吨，单日清运量达到 73 万吨；快递业务量达到 1083 亿件，单日送件量达到 2.97 亿件。为了应对如此大规模的运输需求，城市环境的运输车辆需求随之增加。

由于城市环境下人口密度大，规划复杂，社区结构多样，以及交通状况多样等特征，运输服务复杂性和不确定性成为了大多数城市物流运输行业的主要关注焦点之一。车辆路径规划问题（Vehicle Routing Problem）应运而生。作为运筹优化领域的经典问题之一，路径规划问题旨在满足业务要求的前提下最大限度地提升车辆行驶路线的合理性和效率。经过近些年的发展，车辆规划问题出现了许多变种模型，包括：限载车辆路径规划（CVRP），带时间窗路径规划（VRPTW），取卸货路径规划（PDVRP），以及周期性车辆路径规划（PVRP）。

当前主流车辆路径规划问题中考虑的许多优化目标多为客观指标，如成本最小化、路程最小化、时长最短化、以及时间窗等。以这些优化指标为目标所搭建出来的模型往往可以求解出效率最高和成本最低的路线。但是，此类模型所求解出来的路线在实际应用层面有时却并不友好。比如在人工调度路径规划时，调度员会关注路径的合理性，包括分区服务，行驶线路方便等主观问题，但对于仅考虑客观优化目标的求解方法所获得的路径规划方案则会可能出现路线反复重复，路线过于复杂，或者车辆之间路径重叠的情况。为了解决此类问题，本文提出视觉合理性（Visual Attractiveness）这一主观优化目标，从应用实施层面出发对与路径的规划进行一定的约束。研究表明，对于视觉上合理的线路会使调度员及其他使用者更容易接受并且在实际应用中使用该路径规划方案。下方图 1.1 给出澳大

利亚怡泉饮料公司配送路线在加入视觉合理性与没有加入视觉合理性的路径规划方案。明显的可以看出图 1.1a)相比图 1.1b)更加合理。

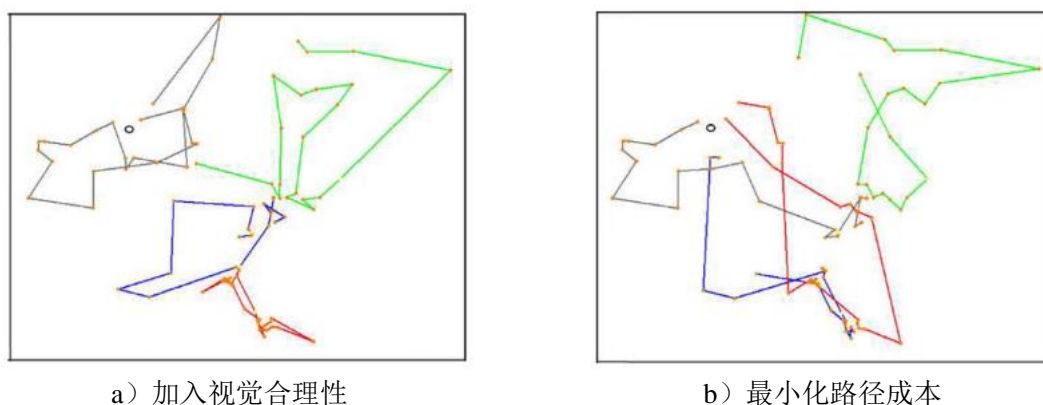


图 1.1 澳大利亚怡泉饮料公司墨尔本城市的配送路线<sup>[2]</sup>

## 1.2 研究意义

在传统人工路线规划中，调度员会基于经验以及司机作业习惯设计路线。通常，调度员会避免路径重叠、交叉以及单车通常会独立负责特定区域。但是由于在传统的车辆路径规划问题中过于注重客观路径长度或路径成本等指标，从而生成违背人工调度逻辑的线路安排。如图 1.2 所示，加入视觉合理性的路径规划方案规避了路径重叠交叉，且路线分区更加明显。因此，本文将着重研究视觉合理性优化在路径规划中的影响力。以城市中的垃圾环卫车路径规划问题为背景，将在综合考虑其运营成本，车辆指标，和垃圾回收站点时间窗等客观指标和约束的前提下，最大限度地优化路径视觉合理性。通过可视化手段去对比传统车辆路径规划问题与加入视觉合理性优化目标的路径规划问题来突显出视觉合理性在路径规划中的重要性与必要性。

同时，随着城市人口的增多和生活节奏的加速，城市内的作业车辆数量将极具增加且工种也变得多样。因城市环境下的复杂的路网特性，将视觉合理性等人的行为因素纳入考量的算法应成为主流（参考图 1.3）。本文采用多目标优化模型来解决车辆路径规划问题，为面向城市内异质车辆的线路优化提供了一定的理论研究基础。

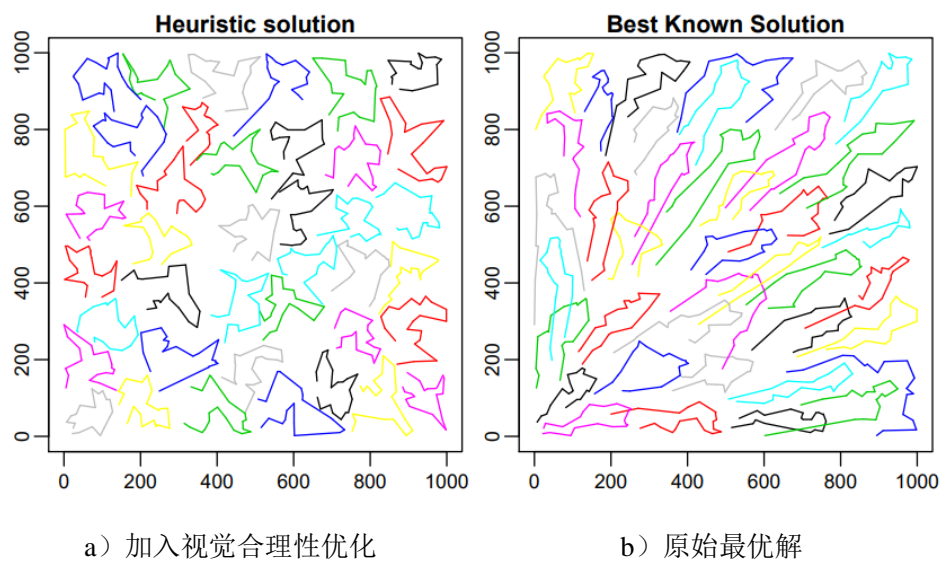


图 1.2 视觉合理性案例二<sup>[2]</sup>

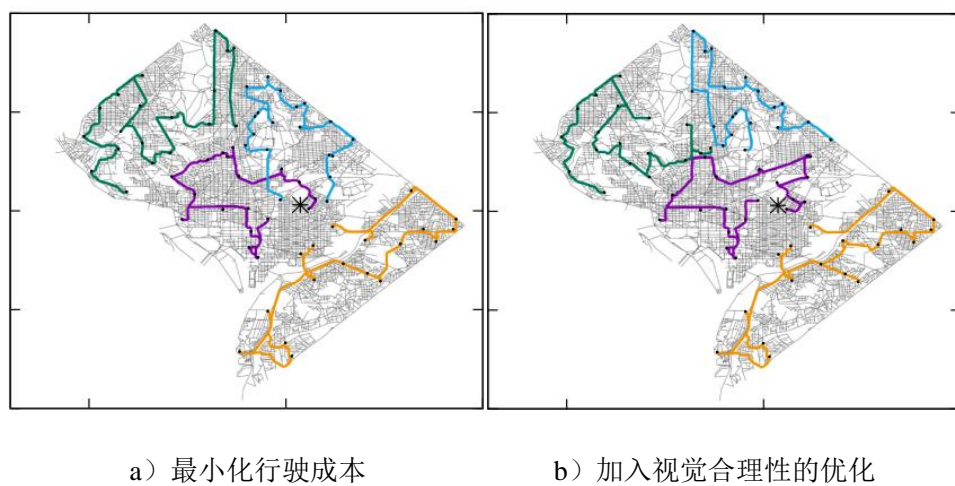


图 1.3 视觉合理性案例三<sup>[1]</sup>

### 1.3 研究内容

经过文献调研，本文基本确定了以下研究内容框架：



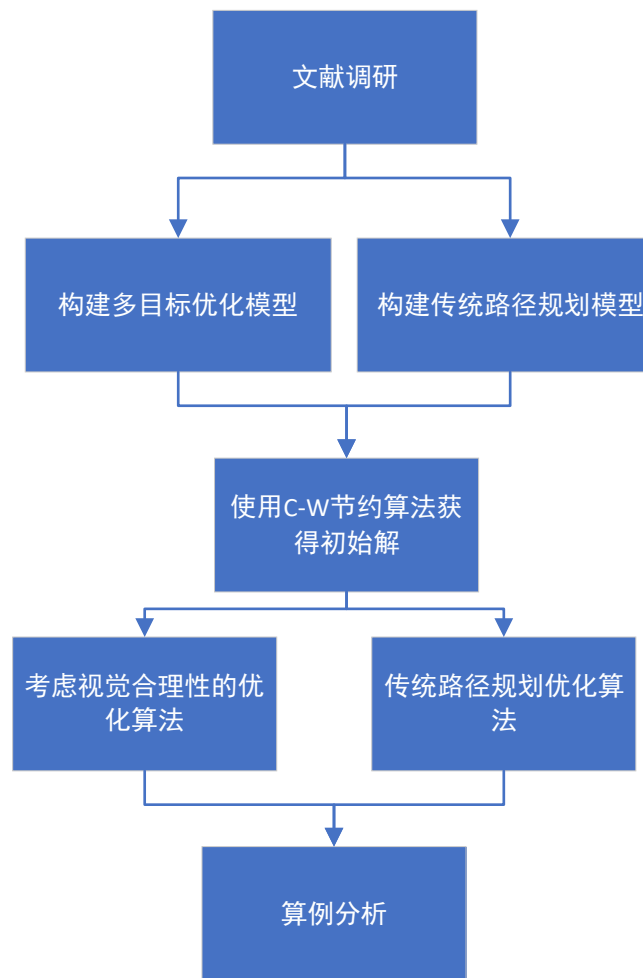


图 1.4 研究内容框架

在综合考虑企业实际运营需求和文献调研后,本文确定了以下几个研究问题:

1. 建立多目标优化数学模型

通过建立运筹学模型将该具体的企业运营问题抽象到数学模型,利用运筹学方法从理论角度重新认识该问题。同时,对提出的视觉合理性考虑进行明确定义以及确定相关量化指标。考虑视觉合理性的数学模型基于基础的VRP模型进行扩展,其在优化目标中需加入紧凑性及路径重叠相关指标,同时在约束中进行定义。

2. 考虑视觉合理性的路径规划问题

此问题为本文的重点研究问题,在原有的路径规划上加入视觉合理性的优化目标,通过数据可视化以及与标准路径规划问题的评估指标对比,验证视觉合理性在路径规划实施层面的重要性。

### 3. 分析最小化成本的路径规划问题

求解最小化路径成本的规划问题为考虑视觉合理性路径规划问题提供参考数据和衡量基准，以验证视觉合理性的必要性。

## 1.4 论文结构

本文将通过文献调研，数学建模，以及算法设计等方法结合工业工程以及运筹学相关理论知识去解决企业业务层面的实际问题，即解决运收作业车辆路径的成本以及合理性优化问题。具体来说，文本将探讨在考虑最优成本的前提下提升城市车辆路径规划合理化。

本文主要包括八个章节的内容：

第一章 引言。介绍本文的研究背景、研究对象和研究意义，明确本论文的内容和结构。

第二章 文献综述。详细介绍现有的相关定义以及研究成果。

第三章 研究问题与研究方法。明确定义本文将要解决的研究问题，对每个问题给出具体的研究方案。

第四章 数学理论模型。构建本文将要用到的所有数学模型，阐释其物理含义。

第五章 算法设计。介绍本文将要使用的启发式算法的具体构建过程以及算法思路。

第六章 案例数据分析。结合实际案例验证数学模型以及启发式算法的可用性，给出算法规划的实际结果，并与算法优化前结果进行比较。

第七章 研究结论。综合数据结果分析，给出本文研究问题的相关结论，基于研究结果，给出合理的运营建议。

第八章 不足与展望。指出本文研究存在的局限性，给出后续研究的方向与建议

## 第2章 国内外研究综述

本章就车辆路径规划问题以及相关问题进行综述。其中包括，车辆路径规划问题背景，现有求解方法，视觉合理性定义，以及考虑视觉合理性相关算法。

### 2.1 路径规划问题

由 Dantzig 和 Ramser 在 1959 年提出的车辆路径问题 (VRP)可以说是最经典的组合优化问题之一。最经典的 VRP 问题，也被称为 Capacitated VRP (CVRP)，是在单一车辆只行驶一条线路、所有车辆指标相同、且只有单一中转节点的背景下在解空间中寻找最优解。旗下的约束包括 1) 所有顾客只能被服务一次；2) 车辆路径的起始点与结束点都为中转节点；以及 3) 车辆的载货量不得超过最大容量。

以这个路径规划模型为基础演变出了许多延伸模型。如异指车队路径规划 Heterogeneous Fleet VRP (HFVRP)，也称为混合车队路径规划 Mixed Fleet VRP。另一主流的延伸是带有时间窗的路径规划 VRP with Time Window (VRPTW)，此模型给定的假设是客户的交付必须在特定时间间隔内完成，该时间间隔因客户而异。当不允许在时间间隔之外交付时，时间窗口被定义为硬时间窗（或严格）。相反，软时间窗则允许在给定时间边界外交付，但会造成一定的惩罚成本。除此之外，近年来的路径规划问题还加入了，包括二氧化氮排放 (Qian and Eglese, 2016)、服务水平 (Vidal et al., 2016)、多中转站点 (Stenger et al., 2013)、视觉合理性 (Rossit et al., 2019) 等多个创新性约束<sup>[14]</sup>。对于不同的应用场景或客户要求，单个路径规划问题可能包含以上提到的多个优化目标，其中不乏相互矛盾的目标。

### 2.2 VRP 求解方法综述

VRP 问题本身为 NP-hard 问题，意味着求解复杂度与难度高。加上 VRP 问题的不同约束变种以及应用算例庞大的规模，寻找最优 VRP 问题求解算法一直是国内外备受追捧的话题。当前已有非常多的 VRP 求解方法被提出，在所有提出的方法中可以大致分为精确算法与近似算法。

精确算法通过建立混合整数线性规划模型，使用线性规划求解方法可以获得精确最优解。当希望获得精确解时，一般可以借鉴分枝定界等经典方法，或使用现代线性规划软件如 Cplex 或 Gurobi 等。精确算法通常会在结构简单、约束较少的小规模问题上才会使用。且大多数精确算法会针对特定的问题而设计特定的求解模型，不具有普适性<sup>[19]</sup>，因此，精确算法在实际中的应用范围非常有限。

现实中更多企业应用层面的 VRP 问题会选择调用近似算法。近似算法主要为一些启发式算法，大致可以分为以下三个大类：

#### (1) 构造启发式算法

构造式启发式算法根据不同的优劣选择标准，会将所有为被分配到线路内的点位进行分配。其中，点位优劣的选择通常跟 VRP 问题的目标函数相关。该方法直接且易懂，并可以快速获得初始解。但其弊端在于获得的解并不稳定，有时会与全局最优解差距过大。具有代表性的构造是启发式算法有 C-W 节约算法和最近邻算法等。因为其不稳定性，构造启发式算法通常不会被单独使用，而是经过改进或与其他算法结合使用。

#### (2) 改进的构造启发式算法

通过调用邻域搜索等方法对构造启发式算法中获得的初始解进行优化。具有代表性的改进构造启发式算法有  $\lambda$ -interchange 算法等。此类算法较容易获得全局最优解，且在许多 VRP 问题研究中已被验证。但其弊端在于随着迭代次数的增长会导致运算时间快速增加。

#### (3) 亚启发式算法

此类算法多为人工智能算法，如遗传算法、粒子群算法、模拟退火、和近些年兴起的神经网络算法等。这些算法相比上两类启发式算法在运行效率上获得了大幅度的提升，同时也更容易求得最优解。但其问题在于可能会因为不同的求解问题背景而陷入局部最优无法逃离。<sup>[20]</sup>

由于精确算法效率低，耗时巨大，目前启发式算法为求解 VRP 的主流方法。

## 2.3 视觉合理性优化目标

视觉合理性（Visual Attractiveness）最早由 Poot et al. 于 2002 年提出<sup>[3]</sup>。在此之前，Poot et al. 使用 ORTEC 路径规划软件所获得的线路被其客户认定为非常差劲。其客户表示，他们强调所谓的差劲与客观传统的路径成本，车辆利用率，行驶路程长度等指标无关，而是体现在一些非标准的可接受度指标上。从客户角度

出发，一个更加视觉上合理的路径规划更加符合逻辑并贴近人工路径规划时的结果，所以这种路径规划结果更加容易从驾驶员和调度员处获得信任，从而接受新的路径规划。在 Rossit et al. 2019 年发表的文章中，其将视觉合理性主要归类成了三类特性<sup>[4]</sup>：

1. 紧凑性：一条单一线路上的需求点位应该相对来说比较靠近；
2. 不重复/不交叉：车辆线路之间应该保持一定的距离，以避免线路交叉；
3. 低复杂度：主要体现在车辆单一线路的结构上（如：单一线路的交叉，线路上的大角度转弯等）。

## 2.4 视觉合理性指标

下方表 2.1 给出相关符号定义：<sup>[2]</sup>

表 2.1 不同线路中心值符号解释

Id.	注释	复杂度
$C_I^1$	线路 I 上所有顾客的几何中间点	$O(n)$
$C_I^2$	位于线路 I 中间的点位	$O(1)$
$C_I^3$	线路 I 上距离集合中间点最近的顾客点位	$O(n)$
$C_I^4$	线路 I 上顾客间距离最小的顾客点位	$O(n^2)$

### 2.4.1 紧凑性

基于空间和地理区域紧凑性，提出  $COMP^a - COMP^f$  六个指标。基于路径与顾客点位距离提出  $PROX^a - PROX^c$  3 个指标<sup>[2][10]</sup>。

表 2.2 不同紧凑性指标

指标	公式	注释
$COMP_I^a$	$\frac{AvgDist_I}{AvgMaxDist_I}$	$AvgDist_I$ 是线路 I 上两个顾客之间的平均距离， $AvgMaxDist_I$ 是所有顾客间距离长度前 20% 的均值。 $COMP_I^a$ 值越大则线路越紧凑

$COMP^b$	$\sum_{I \in K} \sum_{i \in T_I} dist(i, C_I^3)$	值越小越紧凑
$COMP_I^c$	$\sum_{i \in T_I} dist(i, C_I^2)$	值越小越紧凑
$COMP_I^d$	$\frac{\sum_{i \in T_I} dist(i, C_I^1)}{ T_I }$	$ T_I $ 为线路 I 上的顾客数量，值越小越好
$COMP_I^e$	$\frac{\sum_{i \in T_I} \sum_{h \in T_I, h \neq i} dist(i, h)}{ T_I }$	$ T_I $ 为线路 I 上的顾客数量，值越小越好
$COMP^f$	$\frac{\sum_{I \in K} \sum_{i \in T_I} dist(i, C_I^4)}{ T_K }$	$ T_K $ 为线路数量，值越小越好
$PROX_I^a$	$2 \cdot \left(1 - \frac{O'_I}{ T_I }\right) - 1$	$O'_I$ 是线路 I 上到另一线路 J 的中心点的距离 < 到本身线路 I 中心点距离的顾客数量。线路 J 的中心点为 $C_J^1$ 。数值越小越好
$PROX_I^b$	$\frac{O'_I}{ T_I }$	线路 J 的中心点为 $C_J^3$ 。数值越小越好
$PROX^c$	$\sum_{I \in K} O'_I$	线路 J 的中心点为 $C_J^4$ 。数值越小越好

### 2.4.2 非交叉

通过减少路径之间的交叉可以非常有效的提升路线的视觉合理性，本文考虑两个非交叉指标<sup>[2][9]</sup>:

表 2.3 非交叉指标

指标	公式	注释
$X$	$\frac{\sum_{I \in K} X_I}{2}$	$X_I$ 为线路 I 上与其他线路交叉的次数，但不考虑拐角与拐角的交叉
$CH$	$\sum_{I \in K} CH_I$	$CH_I$ 为线路 I 中属于多个凸包的点位数量 使用 Graham 凸包计算公式计算属于多个凸包的用户数量就可对交叉的情况进行定量

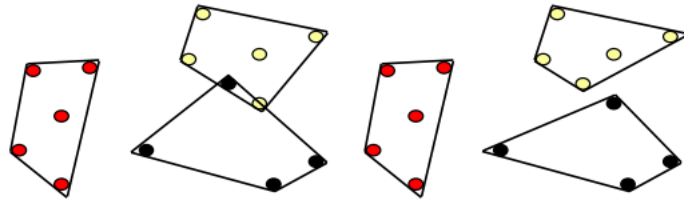


图 2.1 线路凸包的构建与重叠， $CH = 2$ （左）； $CH = 0$ （右）<sup>[5]</sup>

### 2.4.3 低复杂度

低复杂度是从单条线路的路径可驾驶性出发的测量指标，其中包括<sup>[2]</sup>：

表 2.4 低复杂度指标

指标	公式	注释
$IX$	$IX$	$IX$ 为线路 $l$ 上与自身交叉的次数
$CLP_l$	$\frac{\sum_{e \in P_l} \min(\beta_e, \lambda_l - \beta_e)}{\lambda_l}$	其中 $P_l$ 为线路 $l$ 上的点位集合， $\beta_e$ 为在交叉点内的路线长度， $\lambda_l$ 为总路线长度。CLP计算的是交叉部分对于整条线路的占比，数值越小越好
$BE_l$	$\frac{\sum_{i=2}^{ T_l } (\alpha_{i-2,i-1,i})}{ T_l }$	其中 $(\alpha_{i-2,i-1,i})$ 为顾客 $i-2, i-1, i$ 之间的最小角度，数值越小越好。

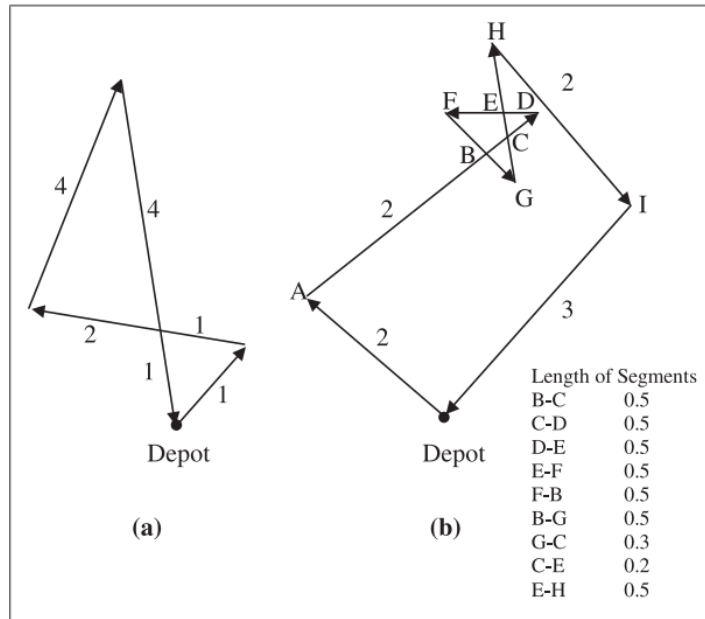


图 2.2 刻画 CLP 概念，为计算线路交叉后较小区域的总里程占比<sup>[6]</sup>

图表 2.2 中(a)的 CLP 值为：

$$= \min \{4 + 4 + 2, 1 + 1 + 1\} / 13 = 0.23$$

图表 2.2 中(b)有三处交叉点，分别在 B、C、以及 E。所以 CLP 值为：

$$\begin{aligned}
 &= \{(BC + CD + DE + EF + FB) + (CD + DE + EF + FB + BG + GC) + (EF \\
 &\quad + FB + BG + GC + CE)\} / 13 \\
 &= \{BC + 2 * CD + 2 * DE + 3 * EF + 3 * FB + 2 * BG + 2 * GC + CE\} / 13 \\
 &= 0.56
 \end{aligned}$$

非常明显图表中的两个路径(a)优于(b)，及证明了 CLP 值越小路径越视觉合理。

## 2.5 视觉合理性约束实现方法

### 2.5.1 聚类法

Rocha et al.以聚类法为基础提出基于聚类法最小化成本及最大化视觉合理性的 VRP 解决方法。下面为其约束模型<sup>[1]</sup>：

表 2.5 符号表



符号	含义
$V$	$V = \{0, 1, \dots, n\}$ 节点, 节点 0 代表中转站, 其余点代表顾客
$a_i$	$a_i > 0$ , 顾客点相应需求量
$\{i, j\}$	$i > j$ , 节点与节点之间的边
$C_{ij}$	节点 $i$ 到节点 $j$ 的路径成本
$X_{ij}$	节点 $i$ 到节点 $j$ 路径的使用次数
$Y_{ij}$	如果点 $i$ 与点 $j$ 被同一辆运输车所服务, 则 $y=1$
$K$	同指标运输车数量
$Q$	运输车辆最大载货量
$f$	$B^{n \times n} \rightarrow \mathbb{R}$ 的聚类指标

$$\text{Min total routing cost} = \sum C_{ij} X_{ij} \quad (2-1)$$

$$\frac{\text{Max}}{\text{Min}} \text{clustering} = f(y) \quad (2-2)$$

s. t.

约束一:

$$x(\delta(\{i\})) = 2 \quad i \in V^+ \quad (2-3)$$

约束二:

$$x(\delta(\{0\})) = 2K \quad (2-4)$$

约束三:

载重约束

约束四:

$$Y_{ij} \geq X_{ij} \quad (2-5)$$

约束五:

$$Y_{ik} - Y_{ij} - Y_{jk} + 1 \geq 0; \quad i, j, k \in V^+, i < j < k \quad (2-6)$$

$$Y_{ij} - Y_{ik} - Y_{jk} + 1 \geq 0; \quad i, j, k \in V^+, i < j < k \quad (2-7)$$

$$Y_{jk} - Y_{ij} - Y_{ik} + 1 \geq 0; \quad i, j, k \in V^+, i < j < k \quad (2-8)$$

式 (2-3) 代表节点唯一性约束, 亦单个顾客节点进入和出去的线路唯一。  
 式 (2-4) 代表中转点访问次数约束。式 (2-5) 代表服务统一性约束, 同一边界连接的两点一定被同一辆车所服务。式 (2-6) 式 (2-7) 式 (2-8) 分别代表线路的点位排序约束。

聚类方法有 3 种:

(1) 最小化周长 (Dmin): 目标中最小化  $D$ , 并且在约束中加入

$$D \geq d_{ij}y_{ij} \quad (2-9)$$

(2) 最小化距离 (MSC): 目标中最小化

$$\sum_{i=1}^n \sum_{i<j}^{n+1} d_{ij}y_{ij} \quad (2-10)$$

(3) 最小平方和 (MSSC): 目标中最小化:

$$\frac{\sum_{i=1}^n \sum_{i<j}^{n+1} d_{ij}y_{ij}}{\sum_{i<j} y_{ij}} \quad (2-11)$$

在算法实现层面, Rocha et al.采用的是 NSGA-II 遗传算法, 通过 ranking 和 crowding distance 对求得最优解。

#### Algorithm 1 NSGA-II framework

---

```

 $P_1 \leftarrow$  initial population
 $t \leftarrow 1$ 
repeat
     $Q_t \leftarrow$  genetic operators on  $P_t$  + local search
     $R_t \leftarrow P_t \cup Q_t$ 
    sort  $R_t$  solution according to ranking
    sort  $R_t$  solution according to crowdingdistance
     $t \leftarrow t + 1$ 
     $P_t \leftarrow \text{selection}(R_t)$ 
until stopping condition satisfied
    
```

图 2.3 NSGA-II 遗传算法的伪代码<sup>[1]</sup>

### 2.5.2 启发式插入算法

Kim et al.在对于带有时间窗的垃圾回收路径规划问题提出了初始解的求解模型。模型结构如下<sup>[7]</sup>:

目标:

最小化使用车辆的数量  
最小化行驶时间  
最大化路径紧凑性  
平衡车辆作业量

约束:

顾客点与仓房的时间窗限制  
车辆最大载货限制  
车辆单日停靠次数限制  
单车线路行驶时间限制  
卸货路线限制  
午休时间

使用的是 Solomon 插入算法的延伸<sup>[8]</sup>。在 Solomon 原先的线路初始[Depot, the seed stop, depot]上再创建一新子线路初始[depot, seed stop, lunch break, the closest disposal facility from the depot, depot] (Step 3)。如果没有 Lunch break 的需求, 可以在放入在之后进行移除。伪代码如下<sup>[7]</sup>:

*Step 0:* 将所有点位标为“未到访”

*Step 1:* If <没有任何剩余“未到访”点位>  
去 Step 6

*Else*

去 Step 2

*Step 2:* 开始新的路线 T; lunch-check = not yet; reference = depot

*Step 3:* If <一个新的子线路 CR 起点可以被选取>

If <T 线路 lunch-check = not yet>

初始化子线路 CR [reference, the seed stop, lunch break, the  
closest disposal facility from the depot, depot]

*Else*

初始化 CR[reference, the seed stop, the closest disposal facility  
from the depot, depot]

*Else*

*去 step 1*

*Step 4: 使用 Solomon 的启发式算在当前子线路 CR 中重复插入“未到访”  
点位直到没有可插入节点  
使用模拟淬火算法 (CROSS 交换和邻域搜索) 对当前子线路 CR 进行优化*

*Step 5: If <子线路 CR 的荷载 < 车辆最大荷载>  
将 CR 中的点位标为“到访”  
将线路 CR 加入到 T 中  
去 step 1*

*Else*

*基于负载约束将 CR 分为两条线路 SR1 & SR2  
将 SR1 中的点位标为“到访”，SR2 中的点位标记为“未到访”  
寻找离 SR1 最近一点最近的下货点  
在 SR1 中加入卸货点位，reference = 卸货点位  
将 SR1 加入 T，更新 T 的 lunch-check 状态  
去 Step 3*

*Step 6: 使用带有 CROSS exchange 邻域搜索的 SA meta-heuristic 优化路径*

*Step 7: 更改卸货点位以及他们在线路中的到访时间以优化路径*

*Step 8: 将 lunch break 删除如果线路可以在午饭之前完成*

*Step 9: 算法结束*

### 2.5.3 邻域搜索

Rossit et al. 提出了结合结合邻域搜索和重构器的视觉合理性优化启发式算法。首先，Rossit 提出了 5 个邻域搜索器<sup>[4]</sup>，分别是：

- (1) One-To-Zero: 此邻域搜索器会选择线路中点到中心点距离超过阈值的点位对其进行重新分配，去寻找另一条线路以满足距离阈值要求
- (2) One-To-One: 此邻域搜索器会选择线路中点到中心点距离超过阈值的点位对其进行重新分配，去寻找另一条线路以满足距离阈值要求
- (3) Merge-And-Rebuild: 此邻域搜索器会寻两条线路当他们其中的点位至少满足 1) 距离对方线路中心点更近 或 2) 线路的凸包相互重叠。邻域搜

索器会将两条条路中的所有点位取消标记，选择两个最远点来作为两条新线路的起始点。

- (4) **Cut-Largest-Path**: 此邻域搜索器会寻找长度异常长的线路来进行切割。异常长度可以定义为线路上顾客点位之间的最长距离。
- (5) **Two-Opt**: 此邻域搜索器在满足约束要求的前提下对两条相互交叉的线路进行替换

在以上的邻域搜索器结果中体现不出明显进步后会开始调用重构器。对于某些需求最低的线路，重构器会尝试将当前线路中的点位分入其他线路中。其中点位插入的优先级为：

- (1) 尝试将当前线路中处于其他凸包中的点位分配到该线路中
- (2) 如不能分配到所在凸包线路中，尝试其他线路，在可以优化整体线路紧凑性的前提下将点位移动到该线路中
- (3) 因为没有其他可插入的线路，则将该点作为新的起始点。

为了避免死循环，最后创建的两条线路将不被重构器考虑在内。

对于每一次线路的重构，都要使用 LKH (Lin-Kernighan-Helsgaun) TSP 启发式算法来进行线路的优化。<sup>[15]</sup>

#### 2.5.4 引导式局部搜索 (Guided Local Search)

局部搜索 (Local Search/LS) 是大部分启发式算法的基础架构。LS 通过对可行域进行迭代寻找其他可行更优解。当所有可行域中的可行解都不优于当前解，则 LS 停止迭代。<sup>[17]</sup> LS 的优势在于可以快速地寻找到优化后的解，但存在的问题则是其容易被困在局部最优解中。为了解决这一问题，不同学者提出了许多不同创新的尝试，包括模拟淬火 (Simulated Annealing)，禁忌搜索 (Tabu Search)，引导式局部搜索 (Guided Local Search)。

引导式局部搜索 (Guided Local Search/GLS)<sup>[17]</sup> 是一种基于局部搜索 (Local Search/LS) 改进而来的元启发式算法。当 LS 被困于局部最优解时，GLS 通过在优化目标中加入对某些特征的惩罚，从而逃离局部最优解。假设 LS 的目标函数为  $g(s)$ ，GLS 则会在此基础上加入新的惩罚目标从而将整体优化目标函数改进为  $h(s)$ ：

$$h(s) = g(s) + \lambda \sum (p_i \times I_i(s)) \quad (2-12)$$

表 2.6 GLS 符号表

符号	含义
$g(s)$	局部搜索目标函数
$h(s)$	引导式局部搜索目标函数
$p_i$	特征 $i$ 的惩罚
$I_i(s)$	为 $(0,1)$ 变量, $\begin{cases} 1 & \text{当特征} i \text{ 存在} \\ 0 & \text{其他} \end{cases}$
$\lambda$	GLS 的超参

GLS 的重点在于选择惩罚特征, 而选择惩罚特征的初衷在于惩罚不被待见的特征或会导致 LS 被困于局部最优解的特征。Hollis, B. L. 与 Green, P. J. 基于 GLS 的算法思路提出了对于 VRP 问题的扩充目标函数<sup>[9]</sup>:

$$\sum_{r \in R} C(r) \left( 1 + \alpha \frac{U(r)}{M(r)} + \beta \sum_{i \in R \setminus r} \frac{H(r, i)}{|S_r|} \right) + \gamma \sum_{r \in R} \sum_{s \in S_r} D(s, G(r)) (1 + \lambda P(S_r, r)) \quad (2-13)$$

公式符号解释如下:

表 2.7 符号表

符号	含义
$R$	线路合集
$C(r)$	为线路 $r$ 的成本
$U(r)$	为未满足最低可接受车辆容量下限的容量
$M(r)$	为最低可接受车辆容量下限
$H(r, i)$	为线路 $r$ 上的顾客位于线路 $i$ 的凸包中的顾客数量
$S_r$	为线路 $r$ 上的顾客点合集
$D(a, b)$	为点 $a$ 至点 $b$ 的欧氏距离
$G(r)$	为线路 $r$ 的中心点位
$P(S_r, r)$	为惩罚函数, 记为顾客 $s$ 被规划至线路 $r$ 的次数

Hollis, B. L. 与 Green, P. J. 在算法实现层面调用两阶段算法, 首先使用插入法生成初始解后使用穷举、GLSAVA (Guided Local Search Algorithm targeting Visual

Attractiveness)、以及 GLSATAD (Guided Local Search Algorithm to minimize Time And Distance) 进行进行优化, 大致如下:

- Step 0:* 先择未被规划的需求点位
- Step 1:* 选择车辆
- Step 2:* 将未被规划的需求点位插入新的路线
  - a:* 最小成本
  - b:* GLSA 辅助插入
  - c:* 摧毁重建插入
- Step 3:* 选择合适的载重车辆
- Step 4:* 继续插入未被规划点位或创造新路线
- Step 5:* 优化路线
  - a:* 穷举法移除与插入
  - b:* 优化路径紧凑性及移除重叠
  - c:* 使用 GLSAVA 进行优化
  - d:* 移除没满足最低载重下限的路线
  - e:* 使用 GLSATAD 优化单个路线

其中关注 *Step 5* 的不同优化步骤。**5a: 穷举法移除与插入**指代遍历所有客户需求点位, 将距离其他线路中心点更近的点位进行移除并重新插入, 直至无法进一步优化, 进而达到优化路径紧凑性的目的。**5b: 优化路径紧凑性及移除重叠**指代将所有路径进行遍历, 检查是否有路径凸包重叠的情况发生, 如有重叠则将会将两条线路一并摧毁。在选择距离最远的两个点作为种子后进行重建。**5c: 使用 GLSAVA 进行优化**指代引导式局部搜索启发式算法, 对整体线路进行优化尝试。**5d: 移除没满足最低载重下限的路线**指代将没被满足最低在中下限的路线进行合并尝试。**5e: 使用 GLSATAD 优化单个路线**进行单个线路的成本优化。

## 2.6 本章小结

经过大量的文献阅读以及现有项目调研, 本文对前述定义的各个研究问题选取各自的研究方法如下:

### 1. 多目标优化数学模型

本文将首先着手搭建垃圾运收车辆路径规划问题的多目标优化模型。通过大量的文献调研以及总结现有视觉合理性量化指标, 在原有的高时效、低成本、以

及路径短的优化模型上进行延伸，加入视觉合理性的优化指标以及约束。以下为本文的模型中的优化目标：

- 1) 低成本：从企业治理角度出发，管理层一定希望将配送服务的成本降到最低以达成利益的最大化。
- 2) 视觉合理性：之前提到过的从视觉合理性出发尽量将路线的规划区贴近人脑的主观意识。现有方法大致分为：
  - a) 紧凑性：将单车的路径停靠点规划的尽量紧凑，这符合“将在附近的点位分到一起”这类主观意识。
  - b) 不交叉：尽量避免车辆路线的交叉，这更符合人类主观逻辑。
  - c) 低复杂度：尽量避免单车路线的蜿蜒曲折和路线交叉等情况，比较符合驾驶员偏爱开直路以便于记忆的逻辑。

而传统约束约束又被分为：

- 1) 时间窗：客户节点会对设两到达时间进行限制。
- 2) 异质车队：车辆规格与限载各不相同。
- 3) 车辆限载：车辆拥有限定的承载量。

## 2. 考虑视觉合理性的路径规划问题

由于考虑视觉合理性的车辆路径规划问题还在研究阶段，现在并没有非常成熟且被广泛认可的算法，所以本文将通过参考并结合不同文献的方法与提出的指标进行视觉合理性的探索。同样，本文将尝试设计启发式算法，但由于视觉合理性的主观性，本文也仅能通过文献调研以及现有求解思路，基于 C-W 节约算法与引导式局部搜索（Guided Local Search）去自行独立开发考虑视觉合理性的启发式算法。

## 3. 最小化成本路径规划问题

最小化路径成本规划问题为基础的 CVRP 问题。本文将基于 C-W 节约算法和局部搜索算法（Local Search）设计并使用启发式算法求得初始解。因为在企业实际应用层面所求解的是规模较大的实际问题，在没有商用求解器的情况下很难直接求解运筹学模型获得精确解。在这种情况下，设计符合要求的启发式算法更加方便快捷的求取近优解是符合实际的做法。



## 第3章 数学模型

本章将所研究的问题进行数学建模，给出符号的定义，展示数学模型的完整性，包括优化目标及各约束条件，并做出解释。

### 3.1 符号定义

本小节在表 3.1 中列明了本章中运筹学模型将会运用到的各种参数和变量符号，并且明确其定义。

如表 3.1 所示，设垃圾点的数量为 $n$ ，垃圾点合集 $N$ 包含所有垃圾点点位。将停车场（所有车辆的统一出发点）标为点位 0，转运站标为 $s$ ，合集 $S$ 包含所有的转运站。垃圾收运车辆的总数目为 $m$ ，垃圾收运车辆合集为 $M$ 。将第 $i$ 个垃圾点位的垃圾量为记为 $q_i$ ，车辆在第 $i$ 个垃圾点位的停留时间记为 $t_i$ ，垃圾点位 $i$ 到垃圾点位 $j$ 的驾车行驶距离记为 $d_{ij}$ 。设垃圾收运车辆 $k$ 的最大载荷为 $c_k$ 。

本问题给出了完整的路径规划方案，确定了每一辆垃圾收运车辆的具体行驶路线，保证服务到所有的垃圾回收需求点位，并且在最优化视觉合理性的前提下使总的车辆行驶里程最短。其中约束包括：每个垃圾点位仅可访问一次，垃圾收运车辆荷载不得超过其最大荷载，总的工作时长不得超过规定时长。

表 3.1 模型参数符号表

符号	含义	符号	含义
$n$	总垃圾点位数量	$tt_i$	收运车辆在点位 $i$ 的停留时间
$N$	垃圾点点位集	$T$	点位时间窗口的时长
0	出发点，即停车场	$d_{ij}$	从点位 $i$ 到点位 $j$ 的距离
$s$	转运站	$c_k$	车辆 $k$ 的核载
$S$	转运站点位集	$q_i$	点位 $i$ 的垃圾量
$v$	垃圾收运车的作业速度	$P_s$	转运站 $s$ 的容量限制
$v_0$	收运车的发车和回场速度	$a_{ijk}^A$	车辆 $k$ 到达 $j$ 的容量
$m$	垃圾收运车的数量	$a_{ijk}^L$	车辆离开 $j$ 的容量
$M$	垃圾收运车集	$H(r, R)$	线路 $r$ 与 $R$ 的凸包重叠面积
$G(k)$	线路 $k$ 的中心点	$D(a, b)$	点 $a$ 与点 $b$ 的欧氏距离

符号	含义	符号	含义
$A(r, R)$	线路 $r$ 与 $R$ 的凸包面积	$F$	非常大的一个正数

根据一般的 VRP 问题数学模型的建立方法，建立如下的决策变量：

$$x_{ijk} = \begin{cases} 1, & \text{车辆 } k \text{ 由点位 } i \text{ 驶向点位 } j \\ 0, & \text{否则} \end{cases}$$

$$t_{ik} \geq 0, \text{ 车辆 } k \text{ 到达点位 } i \text{ 的时间}$$

### 3.2 考虑视觉合理性数学模型

根据表 3.1 中给出的参数与变量符号，运用运筹学理论知识为基础，本节给出如下 VRP 模型，其中包含容量、时间窗、以及不同起终点等约束条件。

式(3-1)：优化目标

$$\min z = \sum_{k=1}^m \sum_{i=0}^{n+s} \sum_{j=0}^{n+s} d_{ij} x_{ijk} \left( 1 + \beta \sum_{z \in M \setminus k} \frac{H(k, z)}{A(k, z)} \right) + \gamma \sum_{k \in M} \sum_{i \in k} D(i, G(k)) \quad (3-1)$$

s. t.

式(3-2)：出场约束

$$\sum_{j=1}^n x_{0jk} = 1, k \in M \quad (3-2)$$

式(3-3)：路径平衡约束

$$\sum_{j=1, i \neq j}^N x_{ijk} = \sum_{j=1, i \neq j}^N x_{jik}, i \in N; k \in M \quad (3-3)$$

式(3-4)：服务次数约定

$$\sum_{j=0}^{n+s} \sum_{k=0}^m x_{ijk} = 1, \quad i \in N; i \neq j \quad (3-4)$$

式(3-5): 回场约束

$$\sum_{i=0}^n x_{ipk} = 1, p = n+1; k \in M \quad (3-5)$$

式(3-6): 限载约束

$$\sum_{j=1}^N \sum_{i=1}^N q_i x_{ijk} \leq c_k, k \in M \quad (3-6)$$

式(3-7): 时间窗约束

$$T_{ik} + \sum_{i=1}^n \sum_{j=1}^{n+s} \frac{x_{ijk} d_{ij}}{v} + tt_i - (1 - x_{ijk}) * F \leq T_{jk}, \quad k \in M \quad (3-7)$$

式(3-8): 时间窗约束

$$T_{0k} + \sum_{j=1}^n \frac{x_{0jk} d_{0j}}{v_0} - (1 - x_{0jk}) * F \leq T_{jk}, \quad k \in M \quad (3-8)$$

式(3-9): 时间窗约束

$$T_{ik} \geq E_i, i \in N; \quad k \in M \quad (3-9)$$

式(3 – 10): 时间窗约束

$$T_{ik} \leq L_i, i \in N; k \in M \quad (3 - 10)$$

其中，式（3-1）为模型的优化目标； $d_{ij}x_{ijk}$ 表示路径距离， $H(k,z)$ 表示线路  $k$  点位中与其他凸包重叠的点位， $\frac{H(k,z)}{\sum x_{ijk}}$ 表示点位重叠点位数量在线路  $k$  中的占比， $D(i,G(k))$ 表示点位  $i$  到线路  $k$  中心点的距离。式（3-2）表示所有车辆必须从停车场出发；式（3-3）保证了路线的平衡，确保连接点位  $i, j$  的线路被同一辆车所服务；式（3-4）表示每个站点只能被服务一次；式（3-5）表示每次执行完路径都回到焚烧站；式（3-6）为载重限制约束；式（3-7）与式（3-8）为获取到达点位的时间，其中式（3-7）通过计算需求点位服务时间与行驶时间从而获得到达每个点位的时间式（3-8）通过计算出场行驶距离而获得到达点位时间；式（3-9）与式（3-10）表示到达时间必须在时间窗限制之间。

## 第4章 算法设计

本章提出考虑视觉合理性的车辆路径规划两阶段算法。第一阶段基于节约算法寻找初始解，第二阶段参考引导式局部搜索启发式算法优化路线整体视觉合理性。本章给出了各算法的具体思路、伪代码，以及算法流程框架。

### 4.1 两阶段算法整体介绍

本文提出两阶段算法。第一阶段基于经典 C-W 节约算法求得初始解。第二阶段参考引导式局部搜索(Guided Local Search)对整体线路的视觉合理性进行优化。其中，第二阶段被分为三个步骤 1) 线路紧凑性优化 2) 线路重叠优化 3) 引导式局部搜索优化 (详见图 4.1)。

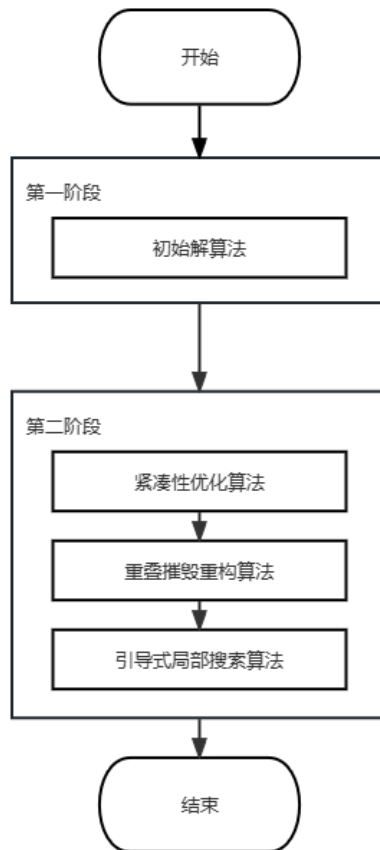


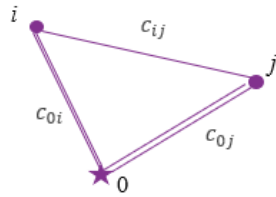
图 4.1 整体算法框架

## 4.2 第一阶段——初始解算法

C-W 节约算法是解决 VRP 问题的经典方法。其思想是根据点位对应的节约值进行降序排列，依次添加点位构造完整路线，使得节约的距离最大。当一辆车的限载约束或者时间窗约束超出限定范围，则更换另一辆车进行线路规划，直至所有点位都被归入线路。

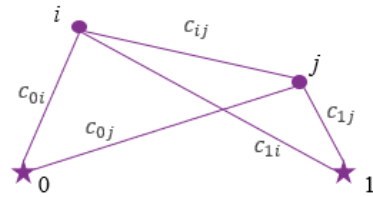
在 C-W 节约算法中，定义并计算节约值是最重要的一部分。同时，由于节约算法要求的特性，其要求单一线路的起始点统一，也就是说车辆从起始点出发需最后回到同一出发点。然而在本文要求解决的问题中，起点为停车场，终点为焚烧厂，线路起始点不同。这导致节约值的计算不再保证非负，如图 4.2 所示。因此，节约算法并不能直接用来解决本文的问题。

• 起点终点相同情况下节约值的计算：



$$\begin{aligned} s(i, j) &= 2c_{0i} + 2c_{0j} - (c_{0i} + c_{0j} + c_{ij}) \\ &= c_{0i} + c_{0j} - c_{ij} > 0 \end{aligned}$$

• 起点终点不相同情况下节约值的计算：



$$\begin{aligned} s(i, j) &= c_{0i} + c_{0j} + c_{1i} + c_{1j} - (c_{0i} + c_{1j} + c_{ij}) \\ &= c_{1i} + c_{0j} - c_{ij} \text{ 正负不确定} \end{aligned}$$

图 4.2 不同情况节约值的计算

沿用节约算法的高效路线生成思路，本文使用点位间距离代替节约值，获得如下算法思路

- (1) 将每个点与起点和终点连接起来，得到  $n$  个点位的完整线路
- (2) 获取所有的点位对  $(i, j)$  的行驶距离  $d(i, j)$ ，对  $d(i, j)$  按照距离进行升序排列。
- (3) 按照  $d(i, j)$  中的顺序，依次考察点位对  $(i, j)$ ，若满足以下条件则将弧  $(i, j)$  插入路线中：
  - a) 点位  $i$  和点位  $j$  不在同一条路线中。
  - b) 点位  $i$  和点位  $j$  与起始点相邻。
- (4) 重复步骤 (3)，直到所有点位对都被遍历。

当点位对间的距离越小，则该点位对就会被越早使用，虽然这种算法可以快速计算出线路解，但其只能给出唯一解。这一弊端会导致后生成的线路质量下降，线路整体分配不够均衡。为解决此问题，本文在上述算法中引入一定的随机性，给出如下改进算法思路：

- (1) 将每个点与起点和终点连接起来，得到  $n$  个点位的完整线路
- (2) 获取所有的点位对  $(i, j)$  的行驶距离  $d(i, j)$ ，对  $d(i, j)$  按照距离进行升序排列。
- (3) 按照  $d(i, j)$  中的顺序，给定一个左偏分布，利用该分布生成的随机数选择点位对  $(i, j)$  进行考察，若满足以下条件则将弧  $(i, j)$  插入路线中：
  - a. 点位  $i$  和点位  $j$  不在同一条路线中。
  - b. 点位  $i$  和点位  $j$  与起始点相邻。
- (4) 重复步骤（3），直到形成完整的路线，计算该方案的总里程，总时间。
- (5) 将新方案与原方案进行比较，若新方案更优则保留新方案。
- (6) 多次重复步骤（3）到步骤（5），迭代获得更优的结果。

该方法伪代码如表 4.1 所示：

表 4.1 随机构造伪代码

引入随机过程的构造算法
<b>Begin</b> Get $d(i, j)$ Get Prob_function Bestsolution = [ ] <b>While</b> stop = False <b>do</b> : <b>Repeat</b> : use Prob_function get a $(i, j)$ solution = combine routes $(i, j)$ <b>If</b> $c_T(\text{solution}) < c_T(\text{Bestsolution})$ <b>then</b> Bestsolution = solution <b>End if</b> stop = evaluation stop rule(true or false) <b>End while</b> <b>Return</b> Bestsolution <b>End</b>

在引入随机性后，该方法将能够在构造阶段，就实现一定程度的路线优化，为后续路线内的优化提供更加高质量的初始解。

### 4.3 第二阶段——视觉合理性优化算法

第二阶段算法按照 1) 路线紧凑性 2) 路线重叠优化 3) 引导式局部搜索进行算法设计。

#### 4.3.1 线路紧凑性优化

紧凑性 (compactness) 是视觉合理性三大指标之一，代指当在同一路径中的需求点位之间距离越近则越符合路径规划逻辑。由此，本小节提出线路紧凑性优化算法，通过缩短同线路点位间距离对线路紧凑性进行优化。

文献中不同的学者提出了近 10 种指标： $COMP^{a\sim f}$  &  $PROX^{a\sim c}$  (见表 2.2)。这些指标分别从地理区域和顾客间距的角度对路径紧凑性进行衡量。本文选择了由 Poot et al. 于 2002 年<sup>[3]</sup>第一次提出视觉合理性概念时所使用的  $COMP^d$  作为定量指标。

$$COMP^d = \frac{\sum_{i \in T_I} dist(i, C_I^1)}{|T_I|} \quad (5-1)$$

其中  $C_I^1$  为线路 I 上所有点位的几何中间点， $|T_I|$  为线路 I 上的点位数量。从物理意义上理解， $COMP^d$  代表的是线路 I 上点位之间的平均距离， $COMP^d$  越小说明线路越紧凑。

本节的算法在第一阶段中所获得初始解的基础上使用穷举获得紧凑性优化最优解。具体算法思路如下：



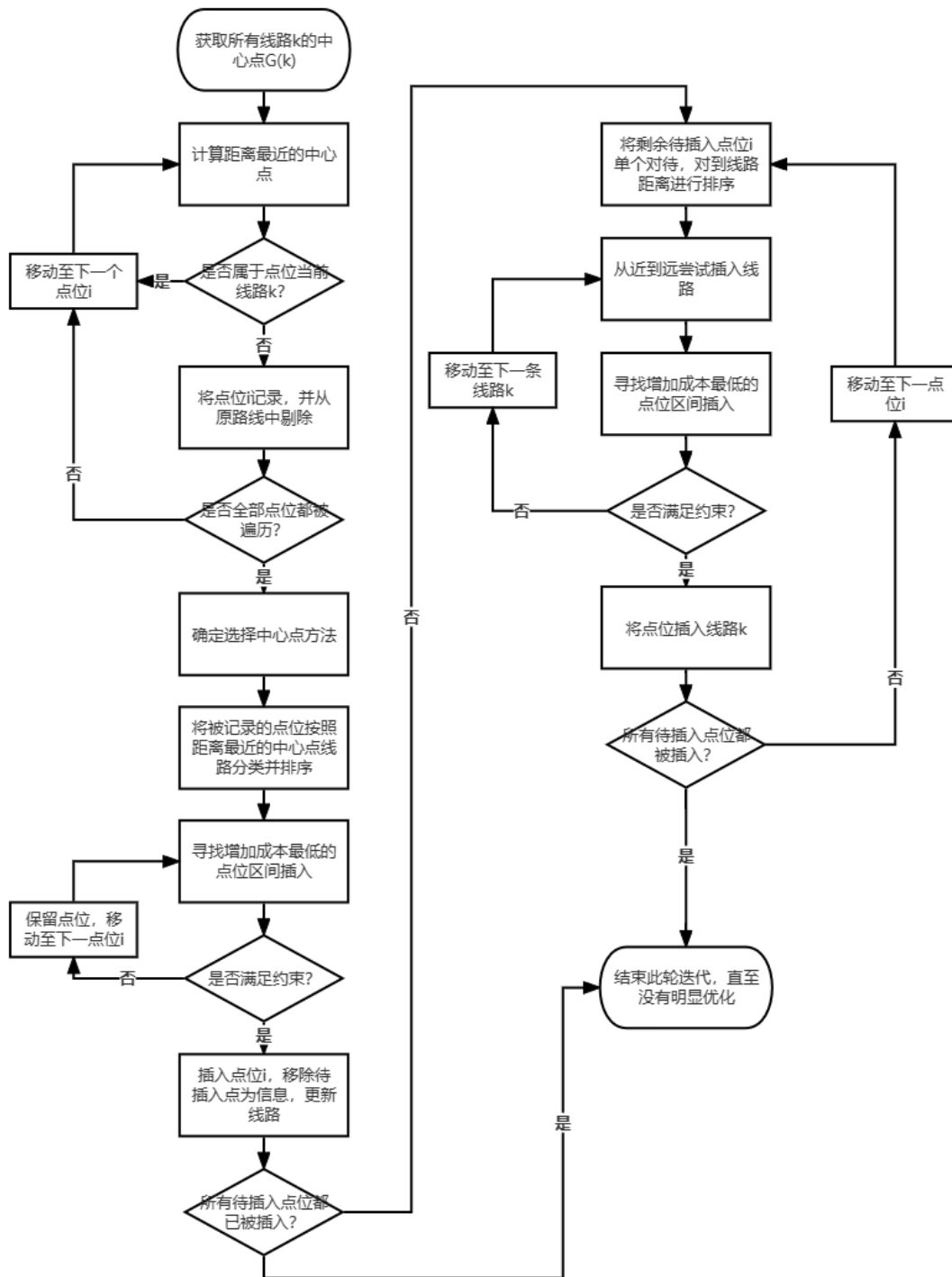


图 4.3 紧凑性优化器算法流程

(1) 计算所有当前线路中心点

- (2) 遍历所有点位, 如点位*i*距离最近的线路中心点非该点位所在线路*k*中心点时将其记录并从原线路*k*中剔除
  - (3) 使用不同计算中心点的方法, 使用确定的中心点 $G(k)$ 为标准对被剔除的点位进行距离排序
  - (4) 在确定约束被满足的情况下, 优先距离最近的点位先插入
  - (5) 对没有被插入到最近线路中的点位*i*单独再次进行计算中心点距离, 并尝试将点位*i*依次插入最近的线路
  - (6) 重复步骤 1-5 直至没有进一步优化
- 伪代码大致如表 4.2 所示:

表 4.2 紧凑性优化伪代码

---

穷举式算法

---

**Begin**

Center = getcenter(Initial Solution)

PointstoInsert = [], InsertDict = {}

**For** *i* in initial solution:

**If**  $\min(d).route \neq i.route$ :

PointstoInsert.append(*i*)

Remove(*i*) from Route

**End if**

Centerused = [Initialcenter, Newcenter, Random]

**While** *j* in PointstoInsert:

InsertDict[min(Getdistance(*j*, centerused).route)].append(*j*)

Sort(InsertDict)

**For** routes in Route:

**For** point in InsertDict[routes]:

Insert(point, routes)

Leftover = Points not inserted

**For** point in Leftover:

Get(min(Getdistance(*j*, Newcenter))

Insert(point, routes)

**Return** Bestsolution

---

---

End

---

#### 4.3.2 重叠摧毁重构器

本小节通过构建路线重叠部分的摧毁与重构，对视觉合理性的第二大特征路径重叠进行优化。为消除车辆路径重叠，本小节提出摧毁重构器(Ruin and Recreate Operator)，具体思路如下：

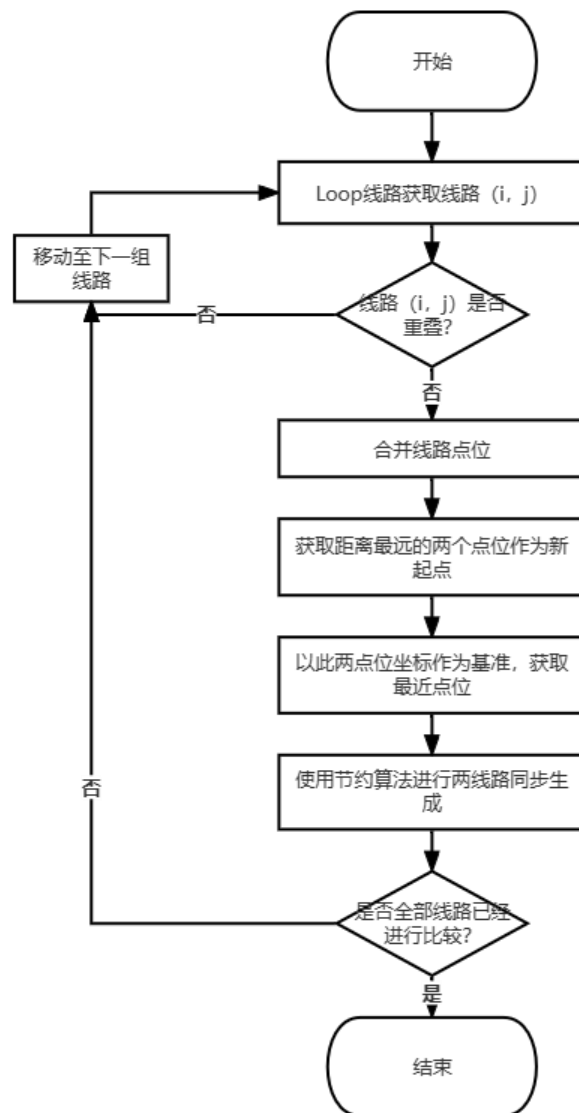


图 4.4 重叠摧毁重构器算法流程

- (1) 对所有线路进行循环比较，获取线路( $i, j$ )
- (2) 如重叠，则摧毁线路( $i, j$ )

- (3) 获得所有点位中距离相距最远的两个点位，将其作为两条新线路的初始点位
  - (4) 对两条线路同时使用节约算法进行重建
    - a) 选择被插入点位时，选择距离初始点位坐标最近的点位
    - b) 寻找成本增长最低的线路区间进行插入
    - c) 检查是否满足约束，是则插入
  - (5) 重复步骤 1-4 直至所有线路被遍历
- 伪代码大致如表 4.3 所示：

表 4.3 线路重叠优化伪代码

---

摧毁重构算法

---

**Begin**

**For** *route i* in Initial Solution:

**For** *route j* in Initial Solution;  $i \neq j$ :

**If** *route i, route j* overlap:

Pointlist = Merge(*route i*, *route j*)

Seed1, Seed2 = Find\_Furthestst\_two\_point(*route i*, *route j*)

Newroute1 = [Seed1]

Newroute2 = [Seed2]

**For** Points in Pointlist:

Find Closest point to Seed1 to Newroute1

**If** feasible: Insert with lowest saving to Newroute1

Find Closest point to Seed2 to Newroute2

**If** feasible: Insert with lowest saving to Newroute2

**End if**

Update route in Bestsolution

**Return** Bestsolution

**End**

---

### 4.3.3 引导式局部搜索

本小节以局部搜索为基础构建包括路径成本和视觉合理性的多目标启发式优化算法。局部搜索通过探索可行解域以寻找更优解，局部搜索的优势在于能快速找到更优解，但劣势在于其容易陷入局部最优解。为此，本文在局部搜

索的基础上加入惩罚值为引导的引导式局部搜索，通过加入惩罚特征以协助局部搜索逃离局部最优。引导式局部搜索思路如下：

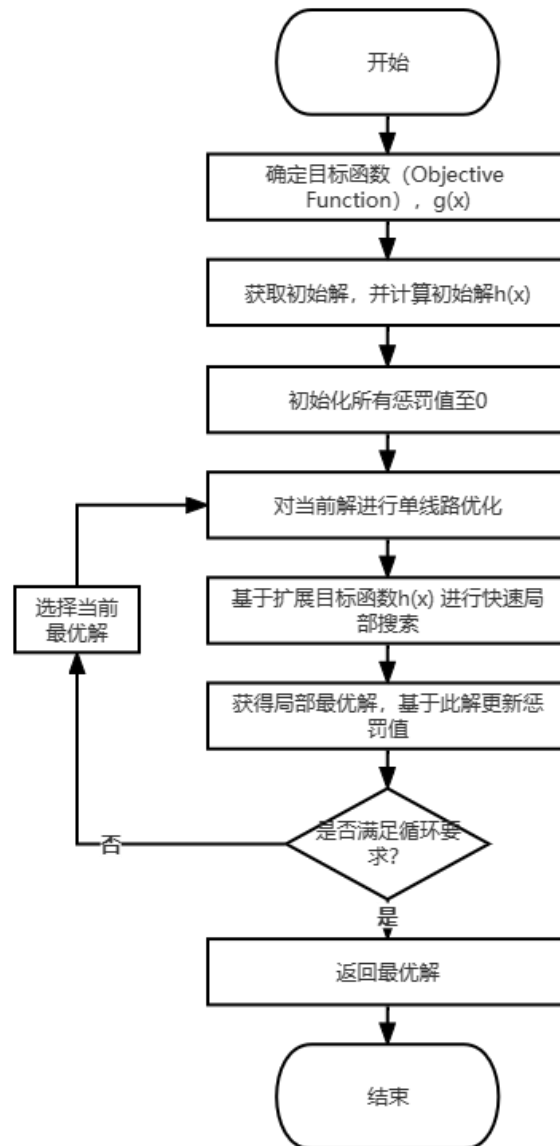


图 4.5 考虑视觉合理性的 GLS 算法流程

下面将对每个部分进行详细的解释：

#### (一) 目标函数与扩展目标函数

目标函数 $g(x)$ 由车辆路径行驶成本和路径重叠构成，用来评估每次局部搜索所生成的局部最优解，当迭代完成则返回拥有最低目标函数值的解。

$$g(x) = \sum_{k=1}^m \sum_{i=0}^{n+s} \sum_{j=0}^{n+s} d_{ij} x_{ijk} \left( 1 + \beta \sum_{z \in M \setminus k} \frac{H(k, z)}{A(k, z)} \right) \quad (5-2)$$

扩展目标函数 $h(x)$ 由目标函数 $g(x)$ 和惩罚成本构成。其中惩罚成本会在每次局部搜索陷入局部最优时更新，惩罚所有不受欢迎的解特征。通过提高惩罚特征的成本，鼓励局部搜索探索可行解时尽量消除不受欢迎特征。

$$h(x) = g(x) + \text{惩罚成本} \quad (5-3)$$

## (二) 惩罚值与惩罚成本

惩罚值与惩罚成本出现在扩展目标函数中，即：

$$\text{惩罚成本} = \gamma \sum_{k \in M} \sum_{i \in k} D(i, G(k)) (1 + \lambda P(i_k, k)) \quad (5-4)$$

在本文所使用的局部搜索所惩罚的特征为紧凑性相关指标。在更新惩罚值时会计算所有线路中点位距离最近的线路，如点位距离最近的线路中心点不是点位所在当前线路则该点位的惩罚值加 1。 $P(i_k, k)$ 为点位 $i$ 在线路 $k$ 的位惩罚值，因为点位可能被分配至任意线路，所以每一条线路会对所有点位进行单独记录。而 $D(i, G(k))$ 为点位 $i$ 到当前线路 $k$ 中心点 $G(k)$ 的距离。 $\gamma$ 则是惩罚函数的一个加权，为超参。

## (三) 单线路内部优化

针对单条车辆行驶线路进行优化，其优化目标为最短路径成本，采用 Two-Opt 方法优化：

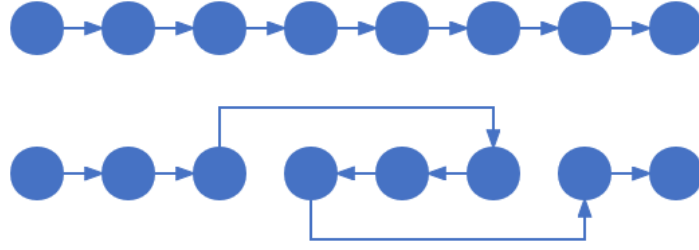


图 4.6 Two-Opt 算法示例

#### (四) 快速局部搜索

如进行全局搜索尝试所有点位的交换，计算 $m$ 条线路共 $n$ 个点位的计算复杂度在 $O\left(m * (m - 1) * \left(\frac{n}{m}\right)^2\right) \sim O(n^2)$ 。可见如果需求点过多，则计算复杂度会迅速增加。为避免大量无意义的计算，算法参考了快速局部搜索，通过经验将算法进行一定约束。在本文中，本文基于经验限制局部搜索仅对相邻的路条路径进行优化尝试。在需求点位增加的情况下可以将复杂度限制在 $O(4 * \frac{n^2}{m})$ ，也就是仅考虑线与路上下左右四条相邻线路进行优化的可行解探索。同时，本文还基于经验加入了限载和时间窗的限制，如点位的加入超出了限载或时间窗的约束则直接忽略。

在这里的快速局部搜索在选择路线优化的方法中分别使用了三种不同的算法：Swap, Cross, Insertion。

##### a) Swap: 线路间点位的交换

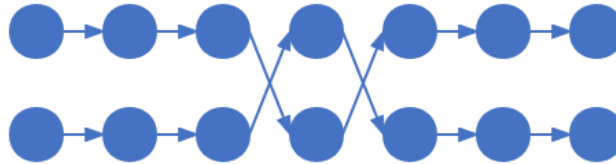


图 4.7 Swap 示例

##### b) Cross: 线路间路线的交换

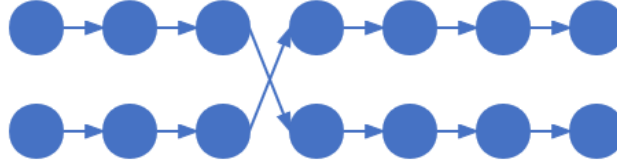


图 4.8 Cross 示例

c) Insertion: 线路间点位的插入

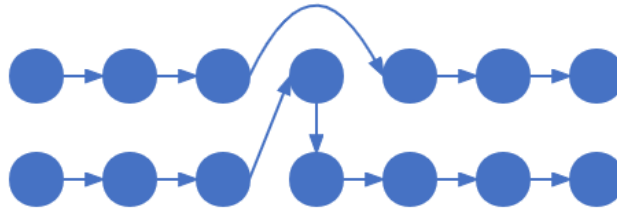


图 4.9 Insertion 示例

该方法总体的伪代码如表 5-3 所示:

表 4.4 引导式局部搜索

元启发式算法
<p><b>Begin</b></p> <p>Bestsolution = Initial Solution</p> <p>BestObjective = obj_func(Initial Solution)</p> <p>Initialize_Penalty</p> <p>SolutionList = []</p> <p><math>Localmin_k</math> = Initial Solution</p> <p><b>Start GLS, While</b> Stopping criteria not met:</p> <p><math>S_k = Localmin_k</math></p> <p><b>Start Local Search, While</b> Stopping criteria not met:</p> <p>Optimize(<math>S_k</math>)</p> <p>Func = Swap, Cross, Insertion</p> <p><b>For</b> route.1, route.2 in <math>S_k</math>:</p> <p><b>Random</b> <math>i</math> in route.1, <math>j</math> in route.2</p>



---

```

 $S_{k+1} = \text{Func}(\text{route}.1, \text{route}.2, i, j)$ 
If  $S_{k+1}$  feasible:
    If  $\text{aug\_obj\_func}(S_{k+1}) > \text{aug\_obj\_func}(S_k)$ 
         $S_k = S_{k+1}$ 
    End If
End If
 $\text{Localmin}_k = S_k$ 
End While
Update_Penalty( $\text{Localmin}_k$ )
SolutionList.append( $\text{Localmin}_k$ )
 $\text{Localmin}_k = \text{Localmin}_{k+1}$ 
End While
Return min(obj_func(SolutionList))

```

---

## 第5章 案例分析

本章首先介绍了城市垃圾运收作业背景，结合位于山西省阳泉市的某垃圾收运企业为研究对象，对上一章中提出的车辆路径规划问题求解算法进行验证。所有算法通过编程实现，编译采用 Python 3.8，编译环境使用 Pycharm Community。

### 5.1 垃圾运收作业背景

本文将以城市垃圾运收作业为企业案例进行车辆路径规划问题研究。当前的城市垃圾收运模式大多是“公交化直运”模式，即垃圾车在路边或易于停靠的路段设置集中垃圾投放点，根据该投放点周边的设施环境配备相应数量的垃圾桶。结合集中投放点的位置，垃圾回收量，站点回收时间等，垃圾车将进行路径规划并且按时依次回收集中投放点中的垃圾后送往焚烧站。

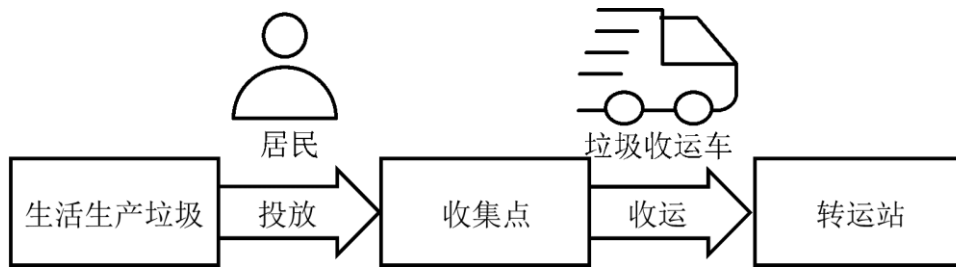


图 5.1 垃圾运收模式示意图<sup>[16]</sup>

目前，环卫公司的垃圾回收车辆路径依然依靠调度人员自身经验来完成路线规划，收运成本高，效率低，同时也无法快速的适应环境改变所带来的线路更新。因此，环卫公司在尝试制定一套标准的路径规划算法，以生成总里程短，作业成本低，车辆利用率高的路径规划方案。

### 5.2 案例背景

本文进行的研究采用的案例来自山西省阳泉市某垃圾收运企业。该项目的优化目标是在满足服务要求的情况下进行环卫车辆路径优化与运营成本优化。其中服务要求约束条件包括服务所有点位，每辆车负责收运的垃圾量之和小于车辆荷

载，在指定时间窗口内完成服务，所有车辆从停车场出发，在焚烧厂结束等。该问题属于带容量约束和时间窗约束的 CVRPTW 问题。按照上述目标生成的现有路径规划方案存在诸多视觉合理性不足的情况。



图 5.2 初期路径规划结果

从图 5.2 中可以明显看出现有路径规划存在重叠和交叉的情况，与基于经验的路径规划存在出入。本章就将基于此算例，在满足所有运营要求的前提下，结合行驶里程，行驶成本等目标，对路径视觉合理性进行优化。

案例的数据方面包含两个算例，每个算例包括需要垃圾收运作业的垃圾点位序号，名称，点位所属街道，点位经纬度坐标，点位的垃圾量，预计作业停留时间，可用时间窗以及车辆通行条件等。其中，车辆通行条件根据路段实际情况分为 3 吨以上车型可通过与 3 吨以下车型可通过，预先处理成立了 0-1 变量便于算法读取，3 吨以上车型可通过设为 1，反之为 0。表 6-1 给出了数据的样例格式。

表 5.1 需求点位数据结构

序号	点位名称	所属路街	经度	纬度
YQCS-284	富力城	南东大街	113. 6188	37. 85098

垃圾量（升/处）	停留时间	允许时间窗	车辆通行条件
903. 1189	2. 25	5:00 - 9:00	1

该环卫企业总共有 4 种不同荷载的垃圾收运车辆，每种荷载的车辆各有 2 辆，共计 8 辆收运车辆。收运车辆作业速度为 15 千米每小时，其他行驶速度为 40 千米每小时。其中荷载对应车辆通行条件，而最大垃圾容积对应车辆限载约束。表 6-2 给出车辆数据格式。

表 5.2 环卫车辆数据结构

车牌号	荷载	最大垃圾容积	最大载重	班次（趟/日）
晋 C24430	8205	44670	10700	2

作业时间	作业速度（km/h）	发车速度（km/h）	回场速度（km/h）
4:30-16:00	15	40	40

5.3 考虑视觉合理性车辆路径规划计算结果与分析

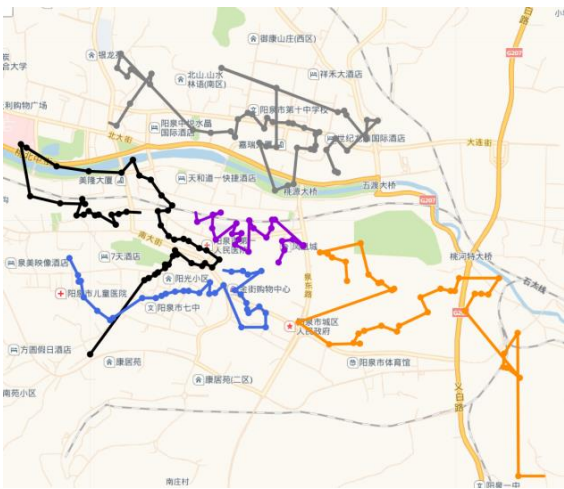
本文采用的求解思路在第四章中进行了详细的解释，此小节将按照提出的两阶段求解思路进行结果计算与分析。

5.3.1 C-W 节约算法计算初始解

算法的第一阶段使用 Clark-Wright 节约算法求解初始解，以下为初始解的可视化地图：



a) 算例一点位初始解



b) 算例二点位初始解

图 5.3 C-W 节约算法初始解

从图 5.3 种可以直观感受到整体路径规划的紧凑性低，路线重叠，以及路线复杂等问题。在初始解中多次的出现了线路交叉，单一线路覆盖面积过大，以及行驶线路过于复杂等情况。

为了方便对比后续阶段优化结果进行比较，表 5.3 给出 4 个路径规划方案关键评估指标分别对应路径成本和视觉合理性相关特性。其中总里程数为方案线路的总距离成本， $COMP^c$ 和 $COMP^d$ 为相对指标以评估线路紧凑性，凸包重叠面积为整体线路间的总重叠面积以评估路线重叠交叉。

表 5.3 初始解关键评估指标

	总里程数（米）	$COMP^c$	$COMP^d$	凸包重叠面积 （平方公里）
算例一	82248.1396	4.7163	4.3204	41.7156
算例二	70377.6931	3.1934	2.9243	0.4518

### 5.3.2 紧凑性优化

本章在紧凑性优化阶段中尝试三种不同的中心点选择方法以及两种计算紧凑性的指标。在紧凑性迭代时会将点位从当前线路移除后再插入，因此线路中心点会在点位移除后受到影响。为此，本章尝试了取出点位后不更新中心点，更新中心点，以及随机选择更新三种方法。而在紧凑性指标的选择上分别选择了：

$$COMP^c = \frac{\sum_{i \in T_I} dist(i, C_I^2)}{|T_I|} \quad (6-1)$$

$$COMP^d = \frac{\sum_{i \in T_I} dist(i, C_I^1)}{|T_I|} \quad (6-2)$$

在代码迭代参数选择上暂定 100 作为迭代次数限制。下面将依次展示对算例一的需求点位使用的三种不同的中心点方法所得到的结果以及可视化。

#### (1) 不更新中心点

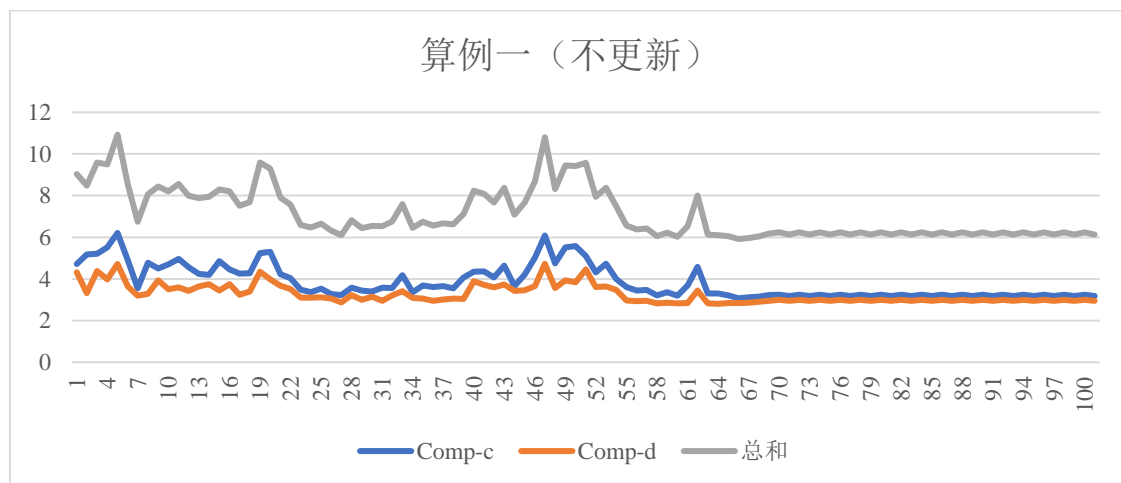
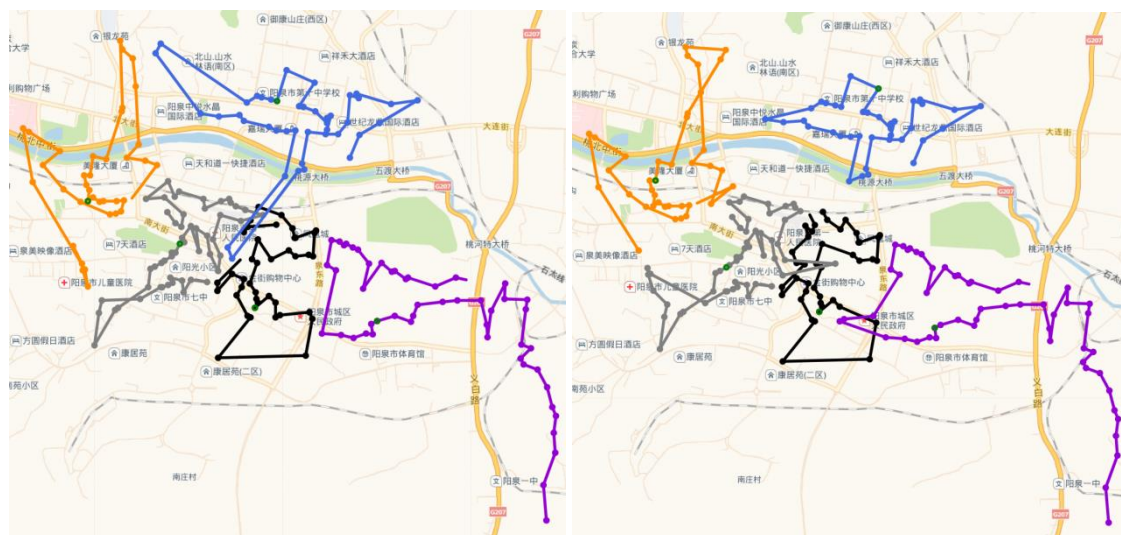


图 5.4 不更新中心点的紧凑性优化收敛



a 最优 $COMP^c$ 以及总和

b 最优 $COMP^d$

图 5.5 不更新中心点方法最优解

## (2) 更新中心点

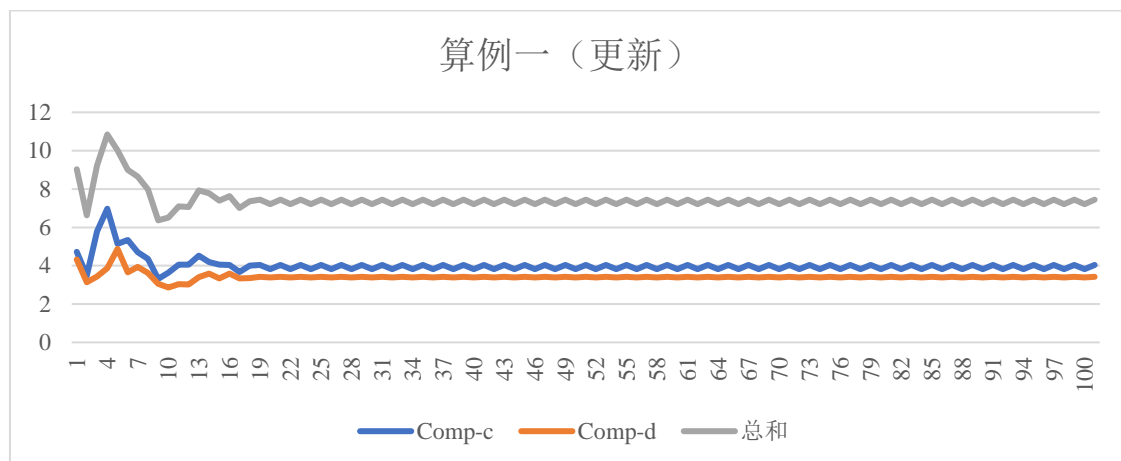


图 5.6 更新中心点的紧凑性优化收敛



a 最优 $COMP^c$ 以及总和

b 最优 $COMP^d$

图 5.7 更新中心点方法最优解

### (3) 随机更新中心点



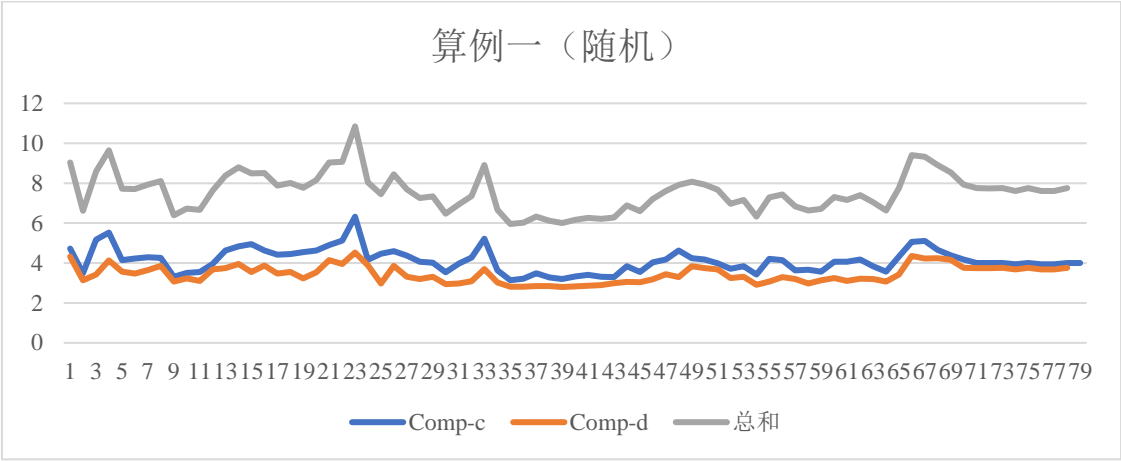


图 5.8 随机更新中心点方法的紧凑性优化收敛

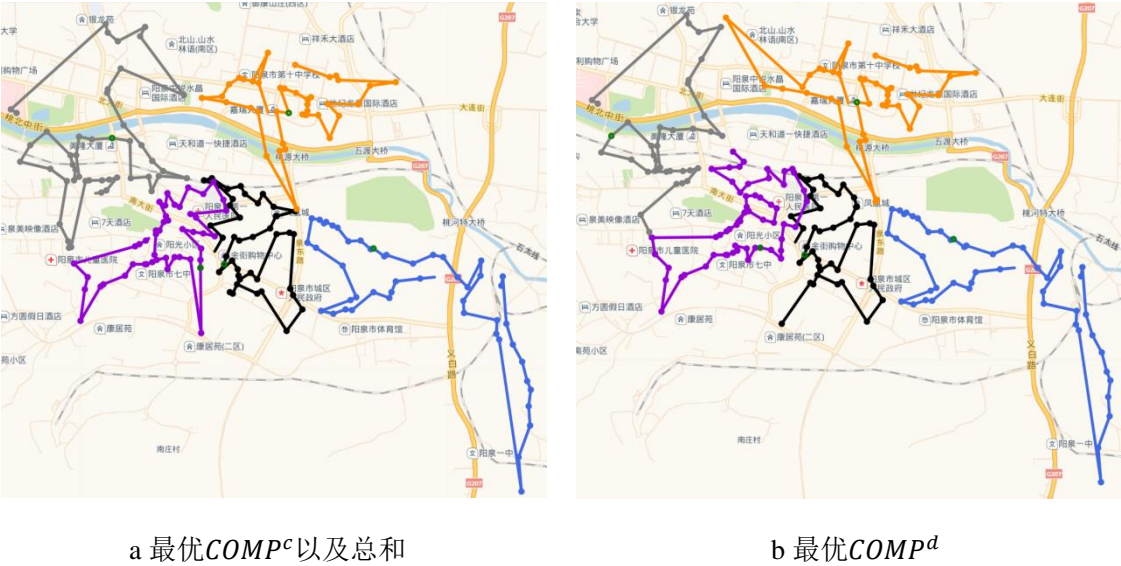


图 5.9 案例一随机更新中心点方法最优解

紧凑性优化结果关键指标如下：

表 5.4 紧凑性优化结果关键指标对比

算例	中心点选择方法	指标	最优计算结果
算例一	不更新	$COMP^c$	3.0716
		$COMP^d$	2.8062
		总和	5.9169
	更新	$COMP^c$	3.3149
		$COMP^d$	2.8638



算例	中心点选择方法	指标	最优计算结果
		总和	6.3693
		$COMP^c$	3.1372
	随机	$COMP^d$	2.8162
		总和	5.9553

在以上三种计算方法中，所有都在 100 次迭代中得到了收敛（见图 5.4，图 5.6 更新中心点的紧凑性优化收敛图 5.6，图 5.8），因此  $n = 100$  是非常保守且合理的选择。其次在对比不同中心点选择方法的可视化地图线路时可以直观的从图像中发现最优  $COMP^d$  解相比最优  $COMP^c$  解分区更加明显，重叠部分更少，线路之间更加紧凑，整体路径方案更加合理。因此，本文将选择  $COMP^d$  作为决定性紧凑性指标。

在对比表 5.4 中不同中心点选择方法结果的关键指标后，可以确定最优  $COMP^d$  的数值在不更新中心点的方法中为最优，最后为更新中心点方法中表现最差。因此建立初步结论不更新中心点方法为此阶段最优解决方法。下方给出算例二的路线规划方案，对上述结论进行验证：



图 5.10 算例二不更新中心点方法  $COMP^d$  最优解



图 5.11 算例二更新中心点方法 $COMP^d$ 最优解



图 5.12 算例二随机更新中心点方法 $COMP^d$ 最优解

表 5.5 对算例二点位的紧凑性优化结果指标对比

算例	中心点选择方法	指标	最优计算结果
算例二	不更新	$COMP^c$	2.8809
		$COMP^d$	2.6380

算例	中心点选择方法	指标	最优计算结果
	更新	总和	5.5253
		$COMP^c$	2.9363
		$COMP^d$	2.7009
	随机	总和	5.6625
		$COMP^c$	2.8600
		$COMP^d$	2.6519
	总和	5.5456	

从表 5.5 中列出的案例二相关数据可以验证 $COMP^d$ 的数值在不更新中心点的方法中为最优，结论成立。因此，本章确定使用不更新中心点作为最终中心点选择方法。

下表给出最终选择的路线方案的关键评估指标：

表 5.6 紧凑性优化结果关键评估指标

	总里程数（米）	$COMP^c$	$COMP^d$	凸包重叠面积 （平方公里）
算例一	83496.5251	2.8062	3.2147	1.8737
算例二	73912.1127	2.6380	2.8943	0

对比初始解（见表 5.3 初始解关键评估指标表 5.3）与紧凑性最优解（见表 5.6），紧凑性相关指标都有所降低，紧凑性指标 $COMP^d$ 最多下降 26%。同时，算例一的凸包重叠降低 96%

### 5.3.3 重叠摧毁重构器

本节的将对阳泉市垃圾运收作业算例一与算例二的路径进行重叠优化。

#### (1) 算例一

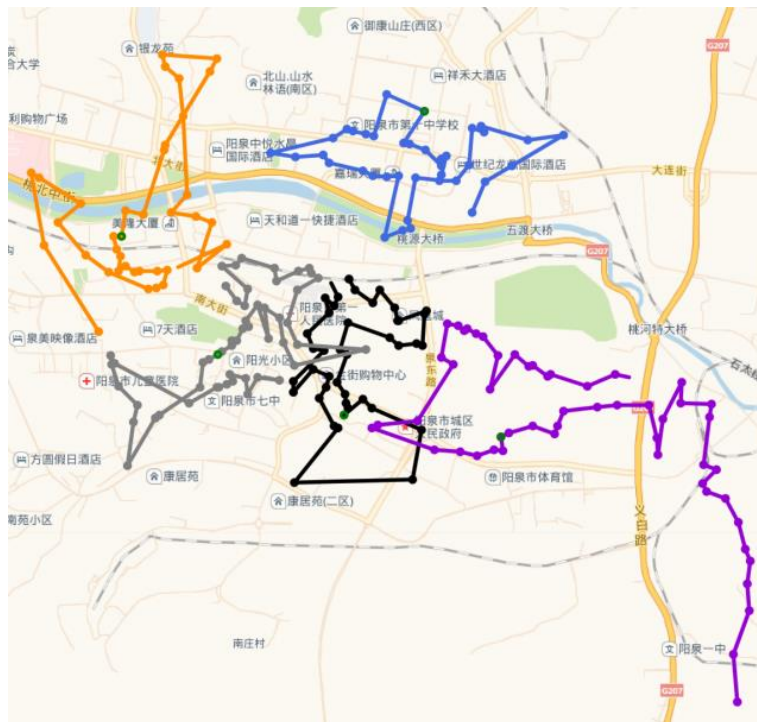


图 5.13 重叠优化前（算例一）

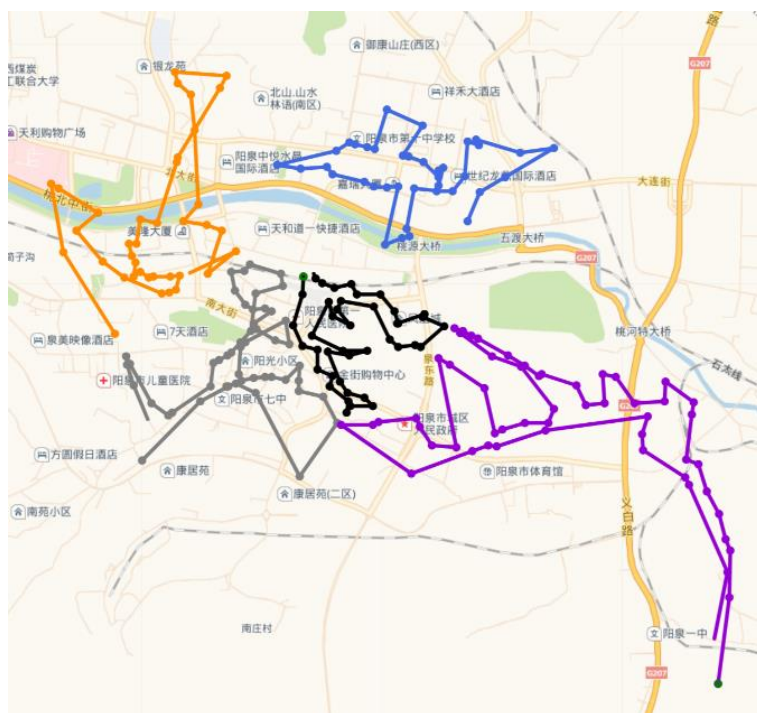


图 5.14 重叠优化后（算例一）

## (2) 算例二

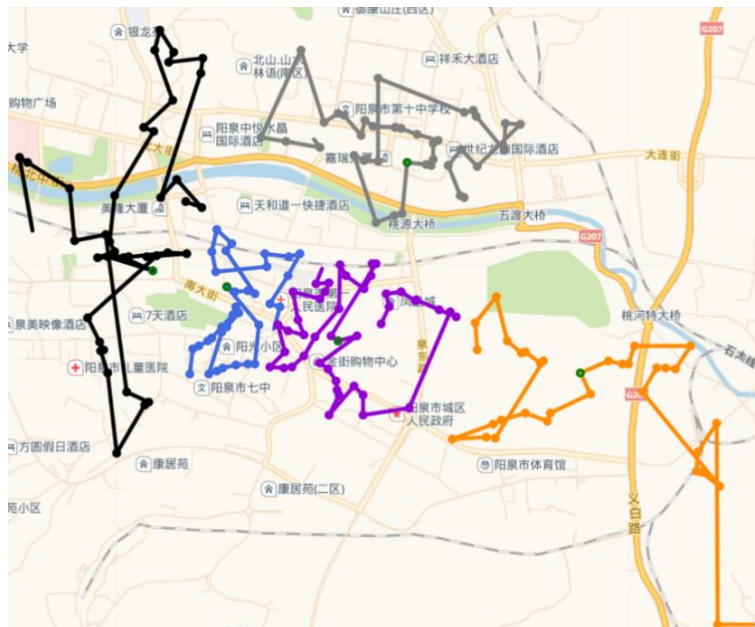


图 5.15 重叠优化前（算例二）



图 5.16 重叠优化后（算例二）

由可视化地图中可以看出算例一的所有路径交叉部分已被消除，算例二优于优化前线路已经没有重叠部分，所以并没有起到优化的效果。下表将给出此阶段优化后的关键评估指标：

表 5.7 重叠重构器优化结果关键评估指标



	总里程数（米）	$COMP^c$	$COMP^d$	凸包重叠面积 （平方公里）
算例一	47833.9315	2.9006	3.3861	0.0199
算例二	73912.1127	2.6380	2.8943	0

对比重叠优化前（见表 5.6）和重叠优化后（见表 5.7），虽然 $COMP^c$ 和 $COMP^d$ 有较小的增长，但算例一线路凸包重叠面积降低 99%，优化效果明显。

### 5.3.4 引导式局部搜索优化算法计算结果

整体路径规划经过上两步的优化已经逐渐趋于合理。本节通过调用考虑总体优化目标的引导局部搜索启发式算法对路径规划方案进行最后的整体优化。通过文献调研以及多次的调参测试，我们将下方局部搜索目标函数 $h(x)$ （式 5-1）中的参数设定为 $\beta = 100$ 、 $\gamma = 0.3$ 、 $\lambda = 3$ 。（见表 5.8）

$$h(x) = \sum_{k=1}^m \sum_{i=0}^{n+s} \sum_{j=0}^{n+s} d_{ij} x_{ijk} \left( 1 + \beta \sum_{z \in M \setminus k} \frac{H(k, z)}{A(k, z)} \right) + \gamma \sum_{k \in M} \sum_{i \in k} D(i, G(k)) (1 + \lambda P(s_k, k)) (5-1)$$

表 5.8 引导式局部搜索参数表

参数	设定值
$\beta$	100
$\gamma$	0.3
$\lambda$	3

下面为本节的结果展示：

#### (1) 算例一

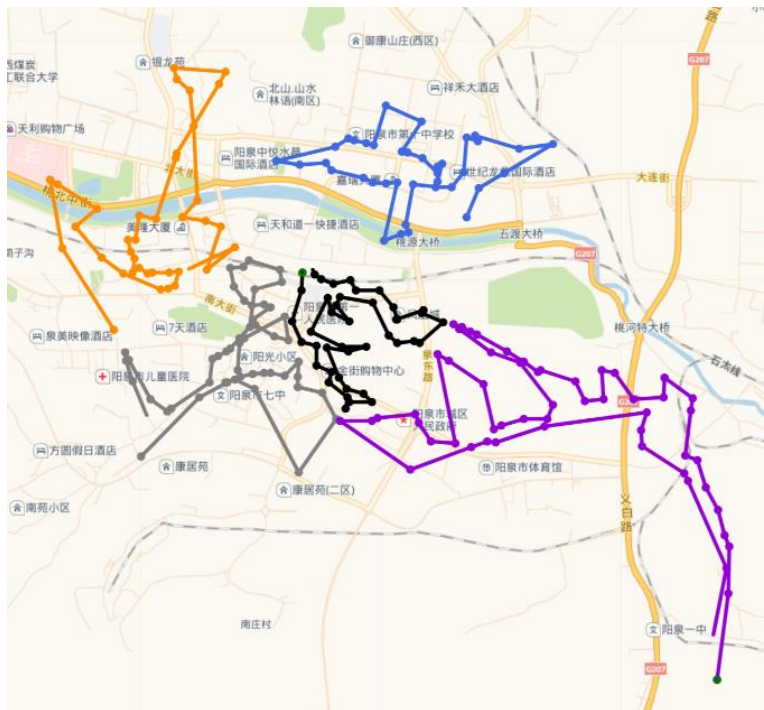


图 5.17 考虑视觉合理性 GLS 优化前（算例一）

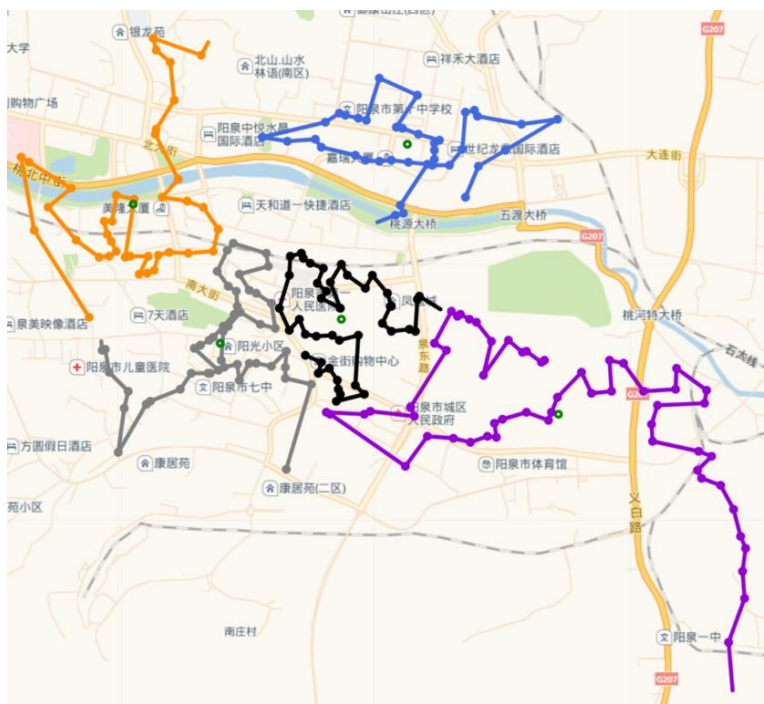


图 5.18 考虑视觉合理性 GLS 优化后（算例一）

## (2) 算例二



考虑视觉合理性 GLS 优化前（算例二）

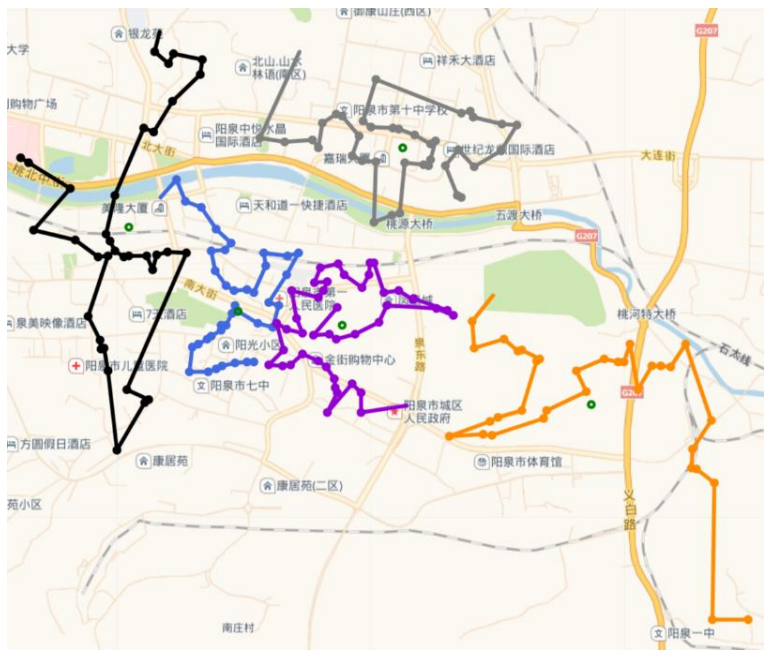


图 5.19 考虑视觉合理性 GLS 优化后（算例二）

从可视化地图中可以明显的看出线路的分区更加明显，已经没有任何线路的交叉，且线路自身的线路规划也更加简洁。

下表将给出此阶段优化后的关键评估指标：

表 5.9 考虑视觉合理性的 GLS 优化结果关键评估指标



	总里程数（米）	$COMP^c$	$COMP^d$	凸包重叠面积 （平方公里）
算例一	38686.9562	2.8921	3.7892	0.1654
算例二	34651.8539	2.8630	3.1303	0.0498

对比引导式局部搜索优化前（见表 5.7）和引导式局部搜索后（见表 5.9）可以看出虽然 $COMP^c$ ， $COMP^d$ ，以及凸包重叠面积有小比例的增加，但线路的总里程数获得了明显的优化。算例一的总行驶里程降低 20%，算例二的总行驶里程降低 53%，优化效果显著。

#### 5.4 仅考虑行驶成本车辆路径规划问题计算结果与分析

本节将通过调用 C-W 节约算法生成初始解，后使用局部搜索（Local Search, LS）生成传统车辆路径规划方案。下方为计算结果的可视化地图和关键评估数据：

##### (1) 算例一



图 5.20 C-W 节约算法初始解（算例一）



图 5.21 最小化路径成本最优解（算例一）

## (2) 算例二

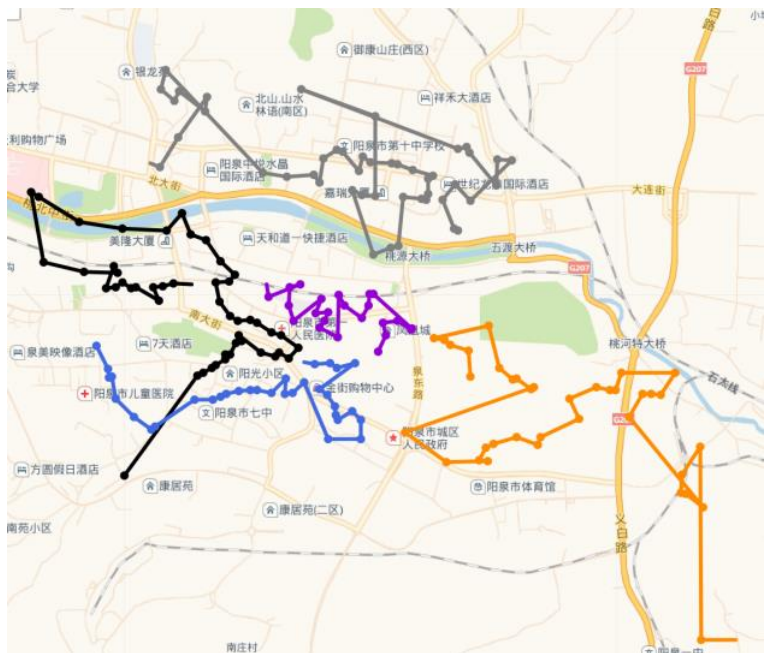


图 5.22 C-W 节约算法初始解（算例二）



图 5.23 最小化路径成本最优解（算例二）

表 5.10 单纯考虑路径成本优化结果的关键评估指标

	总里程数（米）	$COMP^c$	$COMP^d$	凸包重叠面积 （平方公里）
算例一	38016.7630	3.9508	4.2466	10.6364
算例二	33581.8978	3.0792	3.2228	0.8530

通过算例一的可视化地图（图 5.21）可以发现角度不合理性的路径规划，如多出线路交叉，单一线路覆盖范围跨度大等。算例二的可视化地图（图 5.23）较为合理，但由于传统路径规划的合理性对初始解依赖性较高，且算例二初始解本身相对合理，所以算例二参考价值不高。

其次，从关键评估指标表 5.10 可以看出，虽然总里程数相比考虑视觉合理性的结果（表 5.9）有所减少，但其他 $COMP^c$ ， $COMP^d$ ，以及凸包重叠面积都有了明显增加了。其中 $COMP^d$ 最多增加 11%，凸包重叠面积最多增加 98%。因此，本节获得以下结论：

- 1) 方案关键评估指标越低，方案越合理
- 2) 调用考虑视觉合理性路径规划算法所生成的方案更加符合人工规划逻辑

## 5.5 算法验证

为验证算法的普适性，本小节将通过新的算例再次对第四章提出的考虑视觉合理性的算法进行验证。新的算例基于本章前半部分提到的垃圾清运项目，通过打乱需求点位中的坐标以获得新的需求点位，因此依然包含时间窗，限载，以及异质车队等约束。

下面为新算例的结果验证。

a) 初始解

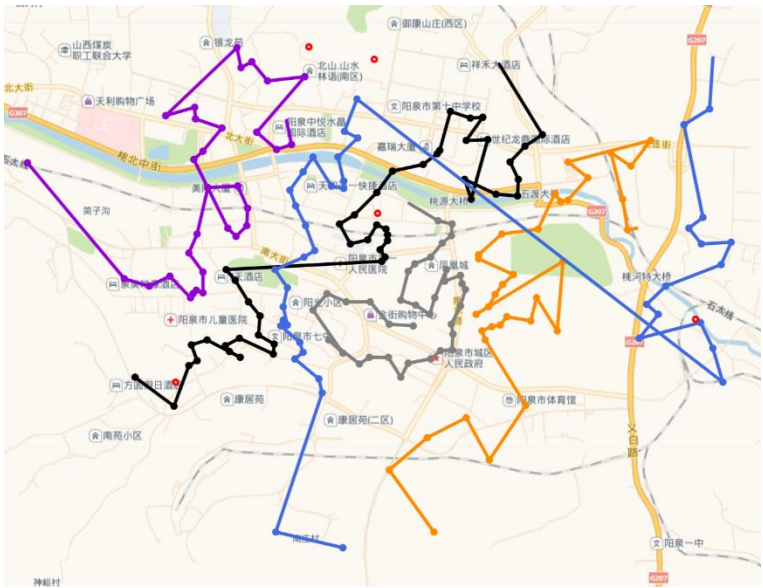


图 5.24 新算例初始解

b) 考虑视觉合理性结果



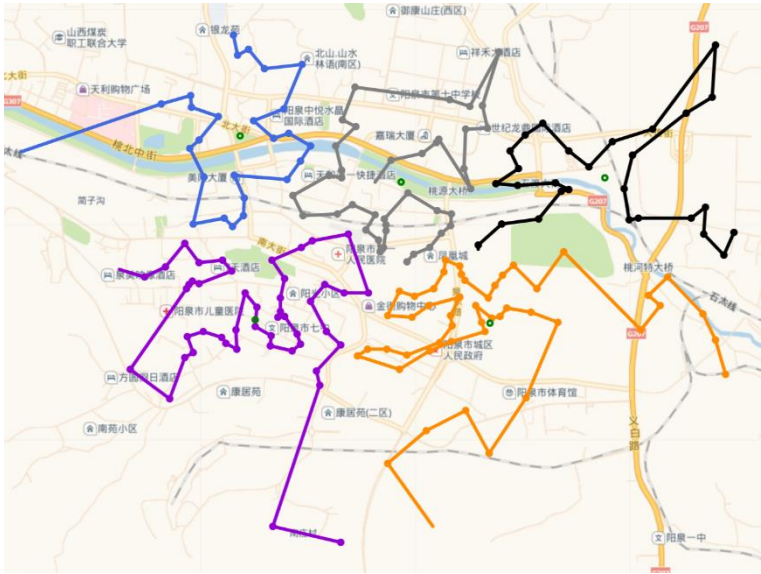


图 5.25 新算例考虑视觉合理性结果

c) 单纯考虑路径成本结果

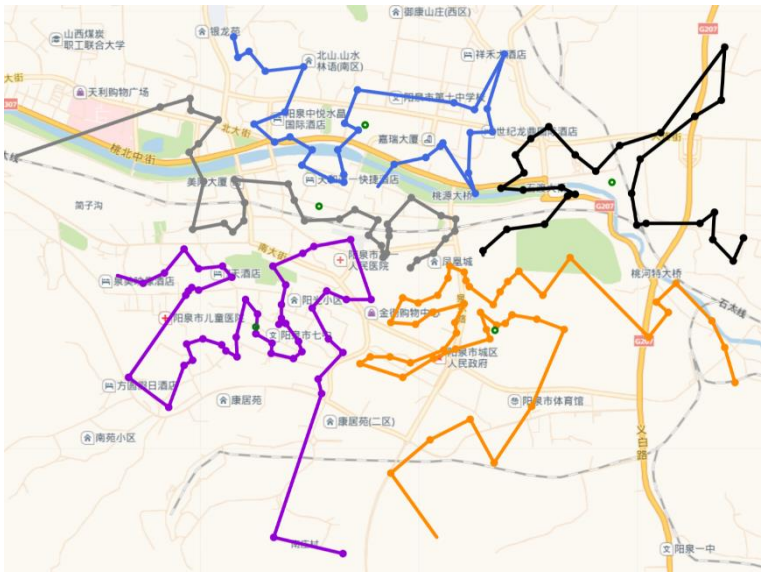


图 5.26 新算例仅考虑路径成本结果

下表为新算例的关键评估指标：

表 5.11 新算例的关键评估指标

总里程数（米）	$COMP^c$	$COMP^d$	凸包重叠面积 （平方公里）
---------	----------	----------	------------------

考虑视觉合理性	47973.0739	3.4922	3.8493	0.0381
考虑成本	47852.2933	3.9092	3.9038	0.0584

首先，从可视化地图（图 5.25）可以直观的看出分区明显，线路之间无交叉，且行驶路径简洁。其次，从关键评估指标（表 5.11）中可以看出除了总行驶里程数外，考虑视觉合理性方案的其他指标相比仅考虑成本的方案有明显的优化。同时需要指出，上半章所使用的算例分析数据集中的点位距离信息为实际行驶距离，而第二个数据集中的点位距离信息则为地理欧氏距离。所以，从新算例中的简洁的线路规划可以侧面印证考虑视觉合理性算法对于路径复杂度的优化。

## 第6章 总结与展望

### 6.1 研究工作总结

本文以城市环卫垃圾作业为背景,对 CVRPTW 车辆路径规划问题进行研究。通过与相关企业的合作项目,从实际应用层面对现有车辆路径规划问题进行优化,提出路线视觉合理性这一创新性考虑。回顾研究历程,本文主要进行了以下工作:

#### (1) 考虑视觉合理性的数学建模

基于传统的路径规划模型,在满足容量约束、时间窗约束、与异质车队等要求的前提下,将视觉合理性指标融入数学模型优化目标,建立完善的混合整数线性规划模,从最终从客观与主观目标对路径规划进行总体优化。

#### (2) 设计总体算法

通过大量文献调研,结合课程学习的相关理论知识,自主设计加入视觉合理性的启发式算法。通过尝试不同的算法框架与关键性指标,经过不断地调参以及算法迭代,最终确定了以 Clark-Wright 节约算法和引导式局部搜索(Guided Local Search)为基础的两阶段启发式算法。

#### (3) 垃圾运收作业路线规划结果分析

本文基于某环卫企业的垃圾收运作业数据对上述算法进行了实例验证。此部分总共使用三个算例,分别为原始数据集的算例一需求点位,原始数据集的算例二需求点位,以及更新后的需求点位。本文通过对比每个算例的传统路径规划方案与考虑视觉合理性的路径规划方案对视觉合理性考虑的必要性进行验证。

### 6.2 研究结论

经过对三个不同的算例分析与对比后对预期结论进行了修改获得以下结论:

“考虑视觉合理性的路径规划方案在不明显影响总行驶里程的前提下,在路径紧凑性,路径交叉,以及路径复杂度等关键指标中都优于传统路径规划方案。其次,相比传统路径规划方案,考虑视觉合理性的路径规划方案在可视化地图中线路分区更加明显,无线路交叉,且行驶路线简洁,符合人工路径规划逻辑。”

数据方面,加入视觉合理性考虑的规划方案相比单纯考虑路径成本的规划方案的在路径紧凑性指标 $COMP^d$ 最多减少了 10.7%,最多去除了所有路线交叉共 5 处,并最多减少 98.4%线路凸包重叠面积。通过将现有算例打乱更新后进行二次

验证，可以确定文本所提出的路径规划算法在不明显增加行驶里程的前提下对各视觉合理性指标都获得了显著的优化。

同时，单纯考虑车辆路径成本的路径规划方案更加依赖于初始解，如初始解存在诸多不合理性，则单纯考虑路径成本的规划方案可能无法对这些进行优化。因此，其生成的最终路线规划方案也会存在很多违背人工线路规划逻辑的不合理之处。相比单纯考虑路径成本的规划方案，加入视觉合理性考虑的规划方案对初始解的依赖性更低，鲁棒性更强，最终生成的方案在满足各项约束的前提下更加合理且符合人工线路规划的逻辑。

最终，结合数据与可视化的两部分结论可以充分证明本文所使用算法的可靠性以及视觉合理性。成功地解决了传统路径规划在企业实际运营及实际应用层面所出现的不足。此外，本文的研究所采用的算法具有普适性，这有便于公司拓展其他服务业务。通过代替人工排程，企业可以更加快速高效的完成垃圾收运路径规划任务，达到降本增效的作用，对企业发展有着深远的意义。

### 6.3 未来研究展望

如今 AI 技术崛起，科技技术进入高速迭代时代，许多企业开始寻求数字化转型。在本项目研究中发现有许多车辆路径规划问题可以基于现代科学技术方法进行改进与优化。

- (1) 城市作业环境多样化复杂且繁忙，还有许多隐形因素未被纳入到路径规划的考虑中。运用人工智能和机器学习算法等方法，可以对城市路径规划问题进行因素归纳与分析，对整体城市路径规划算法进行进一步优化。
- (1) 如今社会对环保及碳排放等话题越加关注，同时车辆运输行业为重碳排放行业。所以为了响应政府号召，应考虑加入相关环境保护的优化目标。
- (2) 可以考虑开发企业运输车队相关程序，将需求点位、车辆、和路况等信息录入。通过调用 VRP 算法自动更新，自动排程，自动规划路径为企业车辆运输服务进行数字化转型达到智能化、标准化、信息化。



## 插图索引

图 1.1 澳大利亚怡泉饮料公司墨尔本城市的配送路线 <sup>[2]</sup> .....	2
图 1.2 视觉合理性案例二 <sup>[2]</sup> .....	3
图 1.3 视觉合理性案例三 <sup>[1]</sup> .....	3
图 1.4 研究内容框架 .....	4
图 2.1 线路凸包的构建与重叠, $CH = 2$ (左); $CH = 0$ (右) <sup>[5]</sup> .....	10
图 2.2 刻画 CLP 概念, 为计算线路交叉后较小区域的总里程占比 <sup>[6]</sup> .....	11
图 2.3 NSGA-II 遗传算法的伪代码 <sup>[1]</sup> .....	13
图 4.1 整体算法框架 .....	24
图 4.2 不同情况节约值的计算 .....	25
图 4.3 紧凑性优化器算法流程 .....	28
图 4.4 重叠摧毁重重构器算法流程 .....	30
图 4.5 考虑视觉合理性的 GLS 算法流程 .....	32
图 4.6 Two-Opt 算法示例 .....	34
图 4.7 Swap 示例 .....	34
图 4.8 Cross 示例 .....	35
图 4.9 Insertion 示例 .....	35
图 5.1 垃圾运收模式示意图 <sup>[16]</sup> .....	37
图 5.2 初期路径规划结果 .....	38
图 5.3 C-W 节约算法初始解 .....	40
图 5.4 不更新中心点的紧凑性优化收敛 .....	41
图 5.5 不更新中心点方法最优解 .....	41
图 5.6 更新中心点的紧凑性优化收敛 .....	42

图 5.7 更新中心点方法最优解 .....	42
图 5.8 随机更新中心点方法的紧凑性优化收敛 .....	43
图 5.9 案例一随机更新中心点方法最优解 .....	43
图 5.10 算例二不更新中心点方法 $COMPd$ 最优解 .....	44
图 5.11 算例二更新中心点方法 $COMPd$ 最优解 .....	45
图 5.12 算例二随机更新中心点方法 $COMPd$ 最优解 .....	45
图 5.13 重叠优化前（算例一） .....	47
图 5.14 重叠优化后（算例一） .....	47
图 5.15 重叠优化前（算例二） .....	48
图 5.16 重叠优化后（算例二） .....	48
图 5.17 考虑视觉合理性 GLS 优化前（算例一） .....	50
图 5.18 考虑视觉合理性 GLS 优化后（算例一） .....	50
图 5.19 考虑视觉合理性 GLS 优化后（算例二） .....	51
图 5.20 C-W 节约算法初始解（算例一） .....	52
图 5.21 最小化路径成本最优解（算例一） .....	53
图 5.22 C-W 节约算法初始解（算例二） .....	53
图 5.23 最小化路径成本最优解（算例二） .....	54
图 5.24 新算例初始解 .....	55
图 5.25 新算例考虑视觉合理性结果 .....	56
图 5.26 新算例仅考虑路径成本结果 .....	56

## 表格索引

表 2.1 不同线路中心值符号解释 .....	8
表 2.2 不同紧凑性指标 .....	8
表 2.3 非交叉指标 .....	9
表 2.4 低复杂度指标 .....	10
表 2.5 符号表 .....	11
表 2.6 GLS 符号表.....	16
表 2.7 符号表 .....	17
表 3.1 多中转站参数符号表 .....	20
表 4.1 随机构造伪代码 .....	26
表 4.2 紧凑性优化伪代码 .....	29
表 4.3 线路重叠优化伪代码 .....	31
表 4.4 引导式局部搜索 .....	35
表 5.1 需求点位数据结构 .....	38
表 5.2 环卫车辆数据结构 .....	39
表 5.3 初始解关键评估指标 .....	40
表 5.4 紧凑性优化结果关键指标对比 .....	43
表 5.5 对算例二点位的紧凑性优化结果指标对比 .....	45
表 5.6 紧凑性优化结果关键评估指标 .....	46
表 5.7 重叠重构器优化结果关键评估指标 .....	48
表 5.8 引导式局部搜索参数表 .....	49
表 5.9 考虑视觉合理性的 GLS 优化结果关键评估指标.....	51
表 5.10 单纯考虑路径成本优化结果的关键评估指标 .....	54
表 5.11 新算例的关键评估指标 .....	56

## 文献参考

- [1] Rocha, D., Aloise, D., Aloise, D. J., & Contardo, C. (2022). Visual attractiveness in vehicle routing via bi-objective optimization. *Computers & Operations Research*, 137. <https://doi.org/10.1016/j.cor.2021.105507>
- [2] Rossit, D. G., Vigo, D., Tohmé, F., & Frutos, M. (2019). Visual attractiveness in routing problems: A review. *Computers & Operations Research*, 103, 13-34. <https://doi.org/10.1016/j.cor.2018.10.012>
- [3] Poot, A., Kant, G., & Wagelmans, A. P. M. (2002). A Savings Based Method for Real-Life Vehicle Routing Problems. *The Journal of the Operational Research Society*, 53(1), 57-68. <http://www.jstor.org/stable/822879>
- [4] Rossit, D., Vigo, D., Tohmé, F., & Frutos, M. (2016). Improving Visual Attractiveness in Capacitated Vehicle Routing Problems: a Heuristic Algorithm. In (pp. 749-756).
- [5] Sahoo, S., Kim, S., Kim, B.-I., Kraas, B., & Popov, A. (2005). Routing Optimization for Waste Management. *Interfaces*, 35(1), 24-36. <https://doi.org/10.1287/inte.1040.0109>
- [6] Lu, Q., & Dessouky, M. M. (2006). A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *European Journal of Operational Research*, 175(2), 672-687. <https://doi.org/10.1016/j.ejor.2005.05.012>
- [7] Kim, B.-I., Kim, S., & Sahoo, S. (2006). Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12), 3624-3642. <https://doi.org/10.1016/j.cor.2005.02.045>
- [8] Solomon, M. M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, 35(2), 254-265. <http://www.jstor.org/stable/170697>
- [9] Hollis, B. L., & Green, P. J. (2012). Real-Life Vehicle Routing with Time Windows for Visual attractiveness and Operational Robustness. *Asia Pac. J. Oper. Res.*, 29.
- [10] Constantino, M., Gouveia, L., Maria, & Nunes, A. C. (2015). The mixed capacitated arc routing problem with non-overlapping routes. *European Journal of Operational Research*, 244(2), 445-456. <https://doi.org/10.1016/j.ejor.2015.01.042>
- [11] Kır, S., Yazgan, H. R., & Tüncel, E. (2017). A novel heuristic algorithm for capacitated vehicle routing problem. *Journal of Industrial Engineering International*, 13(3), 323-330. <https://doi.org/10.1007/s40092-017-0187-9>
- [12] Buhrkal, K., Larsen, A., & Ropke, S. (2012). The Waste Collection Vehicle Routing Problem with Time Windows in a City Logistics Context. *Procedia - Social and Behavioral Sciences*, 39, 241-254. <https://doi.org/10.1016/j.sbspro.2012.03.105>

- [13] Taillard, É., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J.-Y. (1997). A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. *Transportation Science*, 31(2), 170-186. <https://doi.org/10.1287/trsc.31.2.170>
- [14] Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99, 300-313. <https://doi.org/10.1016/j.cie.2015.12.007>
- [15] Helsgaun, K. (2000). An effective implementation of the Lin–Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1), 106-130. [https://doi.org/10.1016/s0377-2217\(99\)00284-2](https://doi.org/10.1016/s0377-2217(99)00284-2)
- [16] Kilby, P., Prosser, P., & Shaw, P. (2002). Guided Local Search for the Vehicle Routing Problem. [https://doi.org/10.1007/978-1-4615-5775-3\\_32](https://doi.org/10.1007/978-1-4615-5775-3_32)
- [17] Voudouris, C., & Tsang, E. P. K. Guided Local Search. In (pp. 185-218). Kluwer Academic Publishers. [https://doi.org/10.1007/0-306-48056-5\\_7](https://doi.org/10.1007/0-306-48056-5_7)
- [18] 向秋雨(2021), 多设施约束下的城市环卫清运车辆路径规划问题
- [19] 方金城 张岐山 物流配送车辆路径问题 (VRP) 算法综述[J]. 沈阳工程学院学报: 自然科学版, 2006, 2(4): 357 – 360
- [20] 庞燕,罗华丽,邢立宁,任腾.车辆路径优化问题及求解方法研究综述[J].控制理论与应用,2019,36(10):1573-1584.

## 致谢

首先，我谨向我的毕业设计导师王晓芳老师表示真诚的感谢，从论文的选题到研究设计，算法框架优化，到最后的結果分析与成果汇报，王老师在每一个环节都给予了大量且全面的指导。每次组会，王老师都会以详细的解答以及耐心的态度对待。老师的这种平易近人和严谨的教学态度协助我顺利的度过了毕设阶段，让我受益匪浅。

同时，我也感谢这四年来清华大学工业工程系对我专业能力以及德智体美劳全面的培养。在这四年中，我收获的学识与能力为我后续的人生道路打下了坚实的基础，成为我迷茫的人生规划中的一盏指路明灯。

感谢这四年来在清华所结交的所有同学，朋友，与老师。这些人在学习和生活上为我提供了许多帮助，也是在这些人的陪伴中让我在清华园里找到了归属感，为大学生活添上了浓墨重彩的一笔。

最后，我要感谢我的父母与家人。感谢父母过去二十多年的栽培和供养，他们一直以来对我的鼓励和支持是我前进的动力。

由衷感谢，感谢清华，感谢工业工程系，感谢身边的所有人。

## 附录 A：外文资料的调研阅读报告

In the traditional Vehicle Routing Problems, the quantitative optimization goals considered are more objective, such as the minimization of transportation costs, travel distances, travel time, etc. In recent years, the new ‘visual attractiveness’ optimization objective have been introduced. Studies have shown enhancing visual in routing plan can effectively facilitate practical implementation and positive collaboration among planning and operational levels in transportation. Although there is no clear definition of the concept of visual attractiveness, but it can be described as planning the routes as logical as possible that resembles how routes were to be planned manually. Many authors have summarized the subjective concept of visual attractiveness into three different characteristics, Rossit, D.G., et al<sup>[5]</sup> described these characteristics as : 1) compact, 2) not overlapping, 3) not complex.

Compactness is one of the most used visual attractiveness measurements. There has been seven ways of defining the compactness of a route: i) similarity to the shape to standard geometric shape; ii) geographical visual compactness; iii) proximity between customers; iv) maximum travel times; v) sum of Euclidean distances; vi) the average and standard deviations of distance between customers; or vii) the perimeter of the zone.

Route overlapping is a more straightforward concept. Poot et al<sup>[3]</sup>. first categorized non-overlapping into two types, total crossing between routes and total crossing within routes. To have a visually attractive route, it is best to keep both values at a low value. This intersection approach was later supplemented by the convex hull approach, where instead of simply considering the routes alone, many researchers take the minimum convex hull formed by the single route. Then by calculating the number of customers who are included in more than one solution, a new indicator has been generated. Needless to say, non-overlapping is a crucial aspect of visual attractiveness because routes who don’t have crossovers are usually considered as more efficient.

Non-complex is related to each route individually and not the overall routing plan. While the number of intra-route crossing is considered as a criteria by some of the researchers, Lu and Dessouky et al<sup>[2]</sup>. do not consider it as a proper quantitative measure to evaluate the crossing level of a route. Instead, they proposed a new measurement, the

crossing length percentage (CLP). The CLP of a route is defined as the sum of the crossed length of all crossing within the single route. Namely, it is the minimum percentage of the total length of crossing. Obviously, the smaller the value of CLP the better the route.

One other mandatory constraint we consider is the time window constraint. It indicates that each customer node has an assigned service timeframe and the service must happen within this frame. Kim et al<sup>[1]</sup>. took an insertion heuristic algorithm approach with a simulated annealing CROSS exchange local search method. The pseudocode is as followed:

```

Step 0:  Mark all stops = unrouted
Step 1:  If there is no unrouted stop,
        Go to step 6
        Else,
        Go to step 2
Step 2:  Start new route T, lunch-check of T = not_yet, reference = depot
Step 3:  If a seed stop for a new sub-route CR can be selected,
        If lunch-check of T = not_yet
        Initialize CR with the sequence (reference, the seed stop, lunch break, the closest
        disposal facility from the depot, depot)
        Else,
        Initialize CR with the sequence (reference, the seed stop, the closest
        disposal facility from the depot, depot)
        Else,
        Go to step 1
Step 4:  Repeat insertion of unrouted stops to the current sub-route CR using
        Solomon's heuristic with time window constraint only until there is no feasi-
        ble unrouted stops
        Improve the sub-route CR by a simulated annealing (SA) meta-heuristic us-
        ing the CROSS exchange local search method
Step 5:  If the required capacity of CR is within the vehicle capacity,
        Mark stops within CR as routed
        Add CR to T
        Go to step 1
        Else,
        Split CR into two sub-routes SR1 & SR2 based on the vehicle capacity constraint
        Mark stops within SR1 as routed and stops within SR2 as unrouted
        Find the closest unloading stop from the last stop in SR1
        Add the unloading stop and complete SR1, reference = the unloading
        stop
        Add the unloading stop and complete SR1, reference = the unloading
        stop
        Add SR1 to T

```



- Update the lunch-check of T
- Go to step 3
- Step 6:* Improve the routes constructed using a SA meta-heuristic with CROSS exchange local search method
- Step 7:* Improve the routes by changing disposal facilities and their positions in the routes
- Step 8:* delete the lunch break from a route if the route can be completed before the break.
- Step 9:* Finish the algorithm

Kim et al. had a lunch break constraint which he denoted in the initialization as lunch-check. At the beginning, an empty route for a vehicle is created, which is marked as T, and the lunch-check is labeled as *not-yet*. Then a sub-route labeled CR is created and a seed stop is selected that is either 1) the farthest stop from the depot or 2) the earliest due time. Kim initialized the sub-route sequence as [depot, the seed stop, lunch break, the closest disposal facility from the depot, depot]. In step 4, customers are repeatedly inserted into the current sub-route without violating the time window constraint. Then the route is optimized using the simulated annealing meta-heuristic CROSS exchange local search method. If the capacity of the optimized sub-route CR is within the capacity of the vehicle, CR is marked as routed and added to T. If the capacity is exceeded, CR is broken into two sub-routes at the max capacity node. The front part of CR is added to T and the latter part is connected after finding the closest unloading stop from the break-up point. After forming all the routes, Kim reapplied SA meta-heuristic with CROSS exchange local search to improve routes.

Rossit et al. built upon this initial solution with 5 local search operators and a reconstructor. 1) One-To-Zero: when the distance between a node to its route center exceeds the distance threshold, this operator seeks to reassign this node to a closer and feasible route. 2) One-To-One: the operator does the same thing as previous operator, but two nodes must interchange. 3) Merge-And-Rebuild: This operator targets two routes where at least one point in each route satisfies i) it is closer to the center of the other route than its own route or ii) the convex hull of the two routes overlap. When this happens, the two route are both unrouted and two most distant customer node are being selected as the seed node of two new routes. 4) Cut-Largest-Path: this operator tries to reorganize the route that is exceptionally long. 5) Two-Opt: This operator tries to find a crossing between routes, and interchange the routes if feasible.

The reconstructor is applied when no improvement appeared in consecutive iterations. The reconstructor operator tries to relocate the nodes of a route with the smallest demand. They relocate with the hierarchy: a) move the node to another route of the convex hull the node is already situated in if feasible; b) move the node to another route that minimizes individual compactness of the route if feasible; or c) if no feasible insertion, create a new route with only one node, making it possible for being selected in the next iteration.

Regardless of the operator selected, each route transformation is applied with the LKH heuristic for beneficial reduction<sup>[4]</sup>.

- [1] Kim, B.-I., Kim, S., & Sahoo, S. (2006). Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12), 3624-3642. <https://doi.org/https://doi.org/10.1016/j.cor.2005.02.045>
- [2] Lu, Q., & Dessouky, M. M. (2006). A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *European Journal of Operational Research*, 175(2), 672-687. <https://doi.org/https://doi.org/10.1016/j.ejor.2005.05.012>
- [3] Poot, A., Kant, G., & Wagelmans, A. P. M. (2002). A Savings Based Method for Real-Life Vehicle Routing Problems. *The Journal of the Operational Research Society*, 53(1), 57-68. <http://www.jstor.org/stable/822879>
- [4] Rocha, D., Aloise, D., Aloise, D. J., & Contardo, C. (2022). Visual attractiveness in vehicle routing via bi-objective optimization. *Computers & Operations Research*, 137. <https://doi.org/10.1016/j.cor.2021.105507>
- [5] Rossit, D. G., Vigo, D., Tohmé, F., & Frutos, M. (2019). Visual attractiveness in routing problems: A review. *Computers & Operations Research*, 103, 13-34. <https://doi.org/10.1016/j.cor.2018.10.012>