



华南理工大学
South China University of Technology

硕士学位论文

考虑时间窗和同时取送货的 车辆路径算法研究

作者姓名	桂梦珂
学科专业	计算机科学与技术
指导教师	刘发贵 教授
所在学院	计算机科学与工程学院
论文提交日期	2021 年 4 月

Research on vehicle routing problem considering time window and simultaneous pickup-delivery

A Dissertation Submitted for the Degree of Master

Candidate: Gui MengKe

Supervisor: Prof. Liu Fagui

South China University of Technology

Guangzhou, China

分类号：TP3

学校代号：10561

学 号：201820132577

华南理工大学硕士学位论文

考虑时间窗和同时取送货的车辆路径算法 研究

作者姓名：桂梦珂

指导教师姓名、职称：刘发贵 教授

申请学位级别：工学硕士

学科专业名称：计算机科学与技术

研究方向：智能计算

论文提交日期：2021 年 04 月 10 日

论文答辩日期：2021 年 6 月 4 日

学位授予单位：华南理工大学

学位授予日期： 年 月 日

答辩委员会成员：

主席：_____陈俊龙_____

委员：_____刘发贵、张星明、贺小箭、高东发_____

摘 要

在电子商务蓬勃发展的大环境下,物流行业已经成为了推动我国经济发展的重要力量,而随着信息化水平的提升,道路网络的不断完善以及人们生活质量的提高,物流配送模式也在不断地改革创新。近年,环境问题和资源再生技术受到了广泛的关注,逆向物流这种取送货一体的物流配送模式正成为众多企业的竞争利器,它带来资源的节约可能意味着经济效益、社会效益和环境效益的共同增加。然而,无论是何种模式的物流服务,居高不下的运输成本仍旧是物流产业面临的最大挑战,因此,优化配送环节有助于物流行业降本增效,也有助于企业加强市场竞争力。基于此背景,本文研究了考虑时间窗和同时取送货的车辆路径问题(Vehicle Routing Problem with Simultaneous Pickup-Delivery and Time Window, VRPSPDTW),主要工作和研究内容如下:

(1) 详细综述了带时间窗的车辆路径问题(Vehicle Routing Problem with Time Window, VPRTW)、考虑同时取送货的车辆路径问题(Vehicle Routing Problem with Simultaneous Pickup and Delivery, VRPSPD)以及它们联合的VRPSPDTW问题的最新研究现状,总结和分析了VRPSPDTW问题现有研究成果中的不足之处。

(2) 根据VRPSPDTW问题组成元素的不同侧重点,对该问题进行了详细分类,总结和对比了解决该问题的常用算法。通过对VRPSPDTW问题特征的分析,构建了该问题的数学优化模型,并且结合了现有研究现状中存在的问题,给出了本文针对该问题求解算法的研究思路。

(3) 设计了一种混合启发式算法(hybrid-Late Acceptance Hill-Climbing Algorithm, h-LAHC)求解VRPSPDTW问题,在生成初始解的步骤中,为了消除基础扫描算法中只根据极角这一单因素进行扫描的不合理性,本文设计了一种具有两个决定性因素的改进扫描生成算法,以生成高质量的初始解。而在优化步骤中,本文指出基础延迟爬山算法(Late Acceptance Hill-Climbing Algorithm, LAHC)的不足,从而提出一种加强型的LAHC算法,并引入强化学习领域的多臂老虎机算法(Multi-Arm Bandit Algorithm, MAB)在搜索深度和搜索广度之间尽可能达到平衡。

(4) 在国际公开的标准数据集上进行了验证实验,结果表明,相比于最新的两种主流算法,本文的求解方案分别在100%和76%的实例上得出了更好或者近似相等的解。特别地,本文的求解方案在时间窗范围相对较窄的场景下具有显著的优越性,而这一场景恰好对应于与人们息息相关的城市配送。此外,本文结合了A物流公司在同城配送上

的实际运营场景，将本文的求解方案成功运用在了 A 公司的实际订单数据上，充分表明了本文求解 VRPSPDTW 问题的方案对现实物流活动具有一定指导意义。

关键词：取送货一体化；带时间窗的车辆路径问题；延迟爬山算法；同时取送货

Abstract

In the booming environment of e-commerce, the logistics industry of China has become an important force to promote economic development. With the continuous improvement of information technology, road network and people's quality of life, the form of logistics distribution is also constantly reforming and innovating. Meanwhile, in recent years, environmental issues and resource regeneration technology have attracted extensive attention. Reverse logistics, the integrated delivery mode of pickup-delivery, is becoming a competitive weapon for many companies. The saving of resources brought by reverse logistics may mean the increase of economic benefits, social benefits and environmental benefits. However, regardless of the form logistics services, the high transportation cost is still the biggest challenge facing the logistics industry. Therefore, optimizing the distribution process will help the logistics industry to reduce costs and increase efficiency, and also help enterprises strengthen their market competitiveness. Based on this background, this paper studies the vehicle routing problem with simultaneous pickup-delivery and time window (VRPSPDTW). The main work and research contents are as follows:

1. The latest research of VPRTW, VRSPD and their combined VRPSPDTW are reviewed in detail, and the deficiencies in the existing research results of VRPSPDTW are summarized and analyzed.

2. According to the different emphases of the constituent elements of VRPSPDTW, the problem is classified in detail, and the common methods for solving the problem are summarized and compared. Through the analysis of the characteristics of VRPSPDTW, the mathematical optimization model of the VRPSPDTW is constructed. Then, this paper combines the existing deficiencies in the current research status, and gives the research ideas of the algorithm for solving this problem.

3. A hybrid heuristic algorithm (h-LAHC) is designed to solve VRPSPDTW. In the step of generating a initial solution, this paper designs an improved sweep generation algorithm with two decisive factors in order to eliminate the irrationality of sweeping only based on the single factor of the polar angle in the basic sweep algorithm. In the optimization step, this paper points out the deficiencies of the basic LAHC algorithm, proposes an enhanced ELAHC algorithm, and introduces the MAB algorithm of the reinforcement learning field to make an attempt to achieve a balance between search depth and search breadth.

4. Simulation experiments are carried out on the internationally public standard data sets. Compared with the latest two mainstream algorithms, the solution in this paper can obtain better

or approximately equal solutions in 100% and 76% of cases respectively. In particular, the solution presented in this paper has significant advantages in a scenario with a relatively narrow time window, and this scenario corresponds to the urban distribution closely related to people's daily life. In addition, this paper combines the actual operation scenario of logistics company A in the city distribution, and successfully applies the proposed solution to the actual order data of company A, which fully shows that the solution of VRPSPDTW in this paper has certain guiding significance for the real logistics activities.

Keywords: Integration of pickup and delivery; Vehicle routing problem with time window; Late acceptance hill-climbing algorithm; Simultaneous pickup and delivery;

目 录

摘 要.....	I
Abstract	III
目 录.....	V
第一章 绪论.....	1
1.1 研究背景和意义.....	1
1.2 国内外研究现状.....	2
1.2.1 VRPTW 问题国内外研究现状.....	3
1.2.2 VRPSPD 问题国内外研究现状.....	5
1.2.3 VRPSPDWTW 问题国内外研究现状.....	6
1.2.4 国内外研究现状分析.....	8
1.3 本文主要创新点.....	9
1.4 本文组织结构.....	9
1.5 本章小结.....	10
第二章 VRPSPDWTW 问题的相关理论和求解算法.....	11
2.1 引言.....	11
2.2 VRPSPDWTW 问题的相关理论.....	11
2.2.1 VRPSPDWTW 问题的定义.....	11
2.2.2 VRPSPDWTW 问题的参与元素.....	11
2.2.3 VRPSPDWTW 问题的分类.....	13
2.3 VRPSPDWTW 问题的求解算法及分析.....	14
2.3.1 精确算法.....	14
2.3.2 传统启发式算法.....	14
2.3.3 智能启发式算法.....	15
2.3.4 混合启发式算法.....	17
2.3.5 求解算法分析.....	17
2.4 本章小结.....	18
第三章 VRPSPDWTW 问题模型构建及算法研究.....	19
3.1 引言.....	19
3.2 基本假设.....	20
3.3 变量的数学表示.....	20

3.4 VRPSPDTW 问题的数学模型.....	21
3.5 VRPSPDTW 问题求解算法研究.....	23
3.6 本章小结	25
第四章 求解 VRPSPDTW 的算法设计.....	26
4.1 引言	26
4.2 求解 VRPSPDTW 的混合启发式算法基本框架.....	26
4.3 初始解生成算法	27
4.4 优化算法设计	28
4.4.1 加强型延迟爬山算法	29
4.4.2 邻域算子	32
4.4.3 智能选择算法	35
4.5 后续优化	37
4.6 本章小结	38
第五章 仿真实验与应用实例	39
5.1 引言	39
5.2 仿真实验	39
5.2.1 实验环境和基准数据集	39
5.2.2 求解参数设置	40
5.2.3 实验结果对比与分析	40
5.2.4 对重要组件的实验分析	44
5.3 应用实例	47
5.2.1 A 公司简介	47
5.2.2 A 公司的同城配送需求	48
5.2.2 应用案例求解	49
5.4 本章小结	51
总结与展望	53
参考文献	55
攻读硕士学位期间取得的研究成果	61
致 谢	63

第一章 绪论

1.1 研究背景和意义

在电子商务蓬勃发展的大环境下，物流业贯穿一二三产业，发展迅猛，已经成为了推动我国经济发展的战略性力量。根据国家发展和改革委员会公布的统计数据显示，2020 年我国物流业虽然受到新冠肺炎的影响，但在第二季度已经实现触底反弹，全年物流总额超过 300.1 万亿元，社会物流总费用为 14.9 万亿元，与 GDP 的比率为 14.7%，而在美国、日本等发达国家中，社会物流总费用在 GDP 中占比均在 9% 以下，反映出我国物流效率相比发达国家仍有较大差距，同时，物流总费用的降低，也直接意味着社会经济效益的显著提高。而在每年的物流总费用中，运输费用的占比超过了 50%，是整体费用的最大支出项，正因如此，降低运输成本是物流业“降本增效”的关键，也直接影响着各大物流企业的盈收效益^[1]。

降低物流运输成本的难题可以抽象为车辆路径问题（Vehicle Routing Problem, VRP），VRP 是物流配送领域的一个经典研究课题，旨在为物流企业提供一组最优化的车辆配送方案，在满足所有客户要求的前提下，使得整个物流活动的成本最小。而随着信息化水平的提升，道路网络的不断完善以及人们生活质量的提高，物流配送形式也在不断地改革创新。除此之外，随着环境问题受到广泛的关注以及资源再生技术的进步，围绕“绿色环保”展开的逆向物流产业正成为众多企业的竞争利器，传统的配送模式也从单一的送货模式逐渐拓展成了取送货一体化的配送模式，实现“资源-产品-可循环资源”的闭环系统。在当今社会的各类活动中，取送货一体化的物流模式非常普遍，比如蓬勃发展的快递行业，为客户配送包裹的同时还提供上门取件的服务，处理客户的寄送需求或者退货需求，除此之外，在食品、电子产品、汽车等工业制造行业，不仅要把产品送到供应点，同时还需要通过逆向的物流活动回收各类包装或者废旧的零部件，以实现资源的循环利用。带时间窗和同时取送货的车辆路径问题（VRPSPDTW）是 VRP 问题在逆向物流领域的重要衍生问题，也是考虑同时取送货的车辆路径问题（VRPSPD）和带时间窗的车辆路径问题（VRPTW）的联合模型。VRPSPD 问题的特点是车辆在同一客户点配送货物的同时也会一并装载上客户需要回收的物品，避免了配送路线的迂回，而 VRPTW 问题的模型对应了更实际的市场需求，因为无论是供应商、销售商还是个人客户，往往希望货物能在预定的时间内送到指定的地点，以便能按照事先规划好的计划来进行生产。VRPSPDTW 问题将时间因素融入到配送中，允许客户预定服务的时间范围，

也要求配送方严格按照指定的时间范围配送或回收货物,这种物流模式更贴近于人们实际的生活购物需求和商务需求。

国内外对于 VRPTW 问题和 VRPSPD 问题的研究已经比较成熟,但对两者联合的 VRPSPDWTW 问题的研究还处于初步阶段。作为经典 NP-Hard 的 VRP 问题的衍生问题,VRPSPDWTW 问题具有更多约束条件,求解难度更大。本文通过分析国内外研究学者对于取送货一体化的 VRPSPDWTW 问题及其相关问题的研究,总结现有研究中的局限,针对该问题目前缺乏简单高效的解决方案,并且缺乏实际应用能力的缺陷,首次提出了一种考虑平衡的混合启发式算法高效求解 VRPSPDWTW 问题,并且将理论成功应用到实际问题中。这不仅为现实生活中带时间窗和取送货一体化的配送过程带来指导意义,也能达到降低运输成本,资源循环利用的目的,在一定程度上能够让经济效益和环境效益共同增加。因此,对于 VRPSPDWTW 问题的研究具有较高的理论和实践意义。

1.2 国内外研究现状

VRP 问题是车辆配送优化的核心问题,也是物流调度领域问题最原始的模型^[2],由于物流配送受到运输企业需求、客户需求和外部环境等多种因素的影响,VRP 问题的多种衍生问题应运而生,使得 VRP 问题及其衍生问题成为运筹学近 60 年来研究最多的课题之一。表 1-1 展示了 2010~2020 这 10 年间,在运筹学核心期刊上发表的关于 VRP 问题及其衍生问题的文章数目,若文章讨论的是两个及多个衍生问题的组合问题,则分别记录。本文着重探讨的 VRPSPDWTW 问题正是 VRPTW 问题和 VRPSPD 问题的联合的更复杂的衍生问题。

表 1-1 近十年在运筹学核心期刊上关于 VRP 问题的文章统计

VRP 衍生问题	描述	文献数目	占比
CVRP	考虑载重的 VRP	218	82.89
VRPTW	考虑时间窗的 VRP	122	46.39
VRPPD	考虑取送货的 VRP	53	20.15
HFVRP	车辆异构的 VRP	45	17.11
MDVRP	考虑多车场的 VRP	29	11.03
GVRP	绿色环保的 VRP	30	11.41
DVRP	动态的 VRP	21	7.98
3D-VRP	三维装载的 VRP	17	6.46
OVRP	开放型的 VRP	12	4.56
VRPSPD	同时取送货的 VRP	10	3.80
TDVRP	时间依赖型的 VRP	10	3.80
MEVRP	多级配送的 VRP	9	3.42

表 1-1 近十年在运筹学核心期刊上关于 VRP 问题的文章统计（续）

MTVRP	大规模运输的 VRP	9	3.42
PVRP	概率型的 VRP	5	1.90
ConVRP	连续型的 VRP	4	1.52
TTRP	拖车运输的 VRP	4	1.52

1.2.1 VRPTW 问题国内外研究现状

VRPTW 问题是车辆路径问题研究最广泛的课题之一，目前现实生活的各类物流调度场景都离不开时间。VRPTW 问题中的客户要求是在预定的时间窗内得到服务，时间窗约定了客户希望被服务的最早开始时间和最晚开始时间，在硬时间窗^[3]的约束下，车辆可以在最早服务时间前到达客户处并等待，但不允许在最晚服务时间之后到达。而在软时间窗的约束下^[4]，车辆可以在客户规定的时间窗前后服务，但有相应的惩罚。在大部分情况下，VRPTW 问题里的时间窗都是硬时间窗，这更符合城市货物运输的场景，毕竟在这个快节奏的时代，延误可能会带来严重的后果。

国外学者 Solomon 最初于 1987 年提出了 VRPTW 问题的优化模型^[5]，通过分析求解 VRPTW 问题的最坏性能，证明了 VRPTW 问题比经典 Np-Hard 的 VRP 问题更复杂。同时，他们公布了 56 个标准数据集，为后来的研究者奠定了理论和数据基础。针对 VRPTW 问题的算法主要有精确算法，传统启发式算法和智能启发式算法，本文将针对这三种方法分别列举其中一部分文献。

精确算法的研究主要在 2010 年以前，研究者为了 VRP 问题提供了列生成算法^[6]、分支定价算法^[7]和动态规划法^[8]，这些精确算法也很好的运用在了 VRPTW 问题上，Qureshi 等人^[9]用列生成算法解决了软时间窗的 VRPTW 问题，N.Azi 等人^[10]和 G. Gutiérrez-Jarpa 等人^[11]分别改进了分支定价算法，并且均在 VRPTW 问题的小规模数据上提供了一组解决方案。精确算法虽然可以求出 VRPTW 问题的唯一精确结果，但是由于问题的 Np-Hard 特性，当客户点的规模变大时，精确算法无法在可接受的时间内得出最优解，这也是精确算法在解决组合优化问题上最大的局限性。

传统启发式算法针对具体问题进行一定步数的可行解构造，虽然不能得到一个最优结果，但是却可以在合理的时间范围内得到可接受的近似解。Solomon^[5]利用节约法、扫描法和插入法解决了 VRPTW，此后运用在 VRPTW 问题上的传统启发式算法基本上也是这几种算法的改进^[12,13]。

智能启发式算法相比于传统启发式算法更具有通用性，不容易被问题本身或者数据实例本身所干扰，因此被广泛应用在 VRPTW 问题和其他 VRP 变种问题上，也是目前

VRP 系列问题在求解方法上的研究重心。Ghoseiri 等人^[14]将 VRPTW 问题视为多目标问题,提出了一种遗传算法,应用帕累托排序过程来寻找非支配解,通过这种方法,可以避免可能的偏向解,达到最优化的目标。Lei 等人^[15]提出了一种自适应大邻域启发式算法 (Adaptive Large Neighborhood Search, ALNS),应用构造启发式方法来提供一个初始解,然后在每次迭代中,大规模的破坏路径和重组路径,每次删除和插入的选择是概率性的,而它们的组合提供了一个广泛的探索空间,实验结果表明该方法能有效解决 VRPTW 问题。Vidal 等人^[16]提出了一种基于多样性控制的混合遗传算法,解决了一系列具有时间特性的 VRP 问题,并引入了一些新的特征来管理时间维度。Yassen 等人^[17]为求解 VRPTW 提出了一种和声智能启发式算法 (Harmony Search Algorithm, HS) 控制算法的进程,用多种不同原理的局部搜索算法作为优化的执行者,并采用一种平衡方法在多种局部搜索算法中进行自适应选择,在标准数据集上的实验结果表明了该算法的高效性。Nalepa 等人^[18]提出了一种自适应的模因算法,弥补了普通模因算法需要频繁调参的不足,在种群进化的过程中自动调节其中的参数。Mehdi 等人^[19]受到 Røpke 等人^[20]提出的自适应大邻域搜索算法 (ALNS) 的启发,通过在 ALNS 算法中加入一种优雅的选择机制来强化 ALNS 的搜索能力,在 Solomon 基准数据集^[5]上进行了实验,结果表明强化后的 ALNS 算法比普通 ALNS 算法在解决 VRPTW 问题上表现得更好。Konstantakopoulos 等人^[21]融合了 VRPTW 问题的多个目标,改进了大邻域搜索算法 (Large Neighborhood Search, LNS),实验结果表明,该算法可以快速解决包含 1000 个客户点的大规模 VRPTW 问题,为现实生活中的客户点分布问题提供了有效的应用方案。

国内对于 VRPTW 问题的研究始于上世纪 90 年代,受国外先进研究经验的启发,国内学者的研究重心在智能启发式算法上,并倾向于将 VRPTW 问题和现实生活中面临的实际场景结合,下面将列举国内学者在 VRPTW 问题上最近的一部分研究。吴斌等人^[22]研究了一种密度峰值聚算法和遗传算法的新型混合算法求解了 VRPTW 问题,并在 9 个数据集上做了实验,验证了该算法比模拟退火算法 (Simulated Annealing Algorithm, SA) 和禁忌搜索算法 (Tabu Search, TS) 两种智能启发式算法的效果更好。盛虎宜等人^[23]将 VRPTW 问题结合生鲜电商配送,考虑了生鲜运输过程中的各种因素,建立了该问题的数学模型并用蚁群算法求解了该问题。于震等人^[24]将 VRPTW 问题和冷链配送结合,并采用人工蜂群算法求解该模型。徐廷学等人^[25]研究了一种量子蚁群算法,将量子计算的概念引入到了蚁群算法中,并使用 2-opt 搜索对优化结果进一步探索,增加获得最优解的概率。蔚帅等人^[26]提出了一种基于密度聚类方法的混合启发式算法,将 SA 和

多邻域下降算法（Variable Neighborhood Decent Algorithm, VND）结合起来对 VRPTW 问题进行求解。马隽等人^[27]在一般的 VRPTW 问题的基础上增加了对装载卸货产生的服务费的研究，建立了符合该应用场景的数学模型，并针对遗传算法的缺陷进行了修正。

1.2.2 VRPSPD 问题国内外研究现状

VRPSPD 问题也是 VRP 问题的一个常见的衍生问题，它结合了正向物流和逆向物流的特点，在车辆配送货物的同时一并装载需要回收的货物。相比于单一的取货或者送货的配送模式，同时取送货的配送模式既能实现资源回收利用，还能避免单一配送过程中车辆回程时的空载问题。

针对 VRPSPD 问题的研究工作由国外的学者展开，Min 等人^[28]通过一个小型图书配送问题的案例初步建立了 VRPSPD 问题的概念，该问题需要在一个中央图书馆和 22 个地区图书馆之间进行图书配送和取书活动。在此之后，许多学者也投入到该问题的研究中。我们将列举其中的一部分。Subramanian 等人^[29-31]致力于使用精确算法来解决这个问题。2010 年，提出了无导向和有导向两种商品流公式，并在多达 200 个客户点的实例上测试了他们的方法。2011 年，提出了一种适用于 VRPSPD 问题的懒惰分支剪切算法，改进了基础分支剪切算法的不足。2013 年，针对 VRPSPD 问题提出了一种分支定价法，在公共的数据集上进行了验证，并将该方法继续运用在了混合取送货的车辆路径问题（Vehicle Routing problem with mixed pickup and delivery, VRPMPD）中。尽管针对 VRPSPD 问题的精确算法已经能够解决多达 200 个客户点的实例，但让人望而却步的运行时间也让研究方向的主流往智能启发式算法上靠拢。Dethloff 等人^[32]是最早使用启发式算法研究 VRPSPD 问题的研究者之一，他指出了 VRPSPD 在逆向物流中的重要性，并提出了一种基于最节约插入方法的算法，也为 VRPSPD 问题提供了一个公共数据集。Nagy 等人^[33]提出了四种基于插入法的启发式方法来求解 VRPSPD 问题，他允许整个搜索过程中出现不可行的解，最后再做修复。在许多组合优化问题中被证明十分强大的禁忌搜索算法（TS），是在 VRPSPD 问题的解决方案中使用频次最高的智能启发式算法。Crispim 和 Brandao^[34]设计了 TS 和 VND 的混合算法，使用这两种启发式算法获得的解是非常多样的，因为它可以探索更广阔的解空间。Wassan 等人^[35]提出了一种反应性禁忌搜索算法解决了 VRPSPD 问题，该算法可以通过判断解的重复情况来自适应地改变禁忌表的大小，实现了搜索强度和多样性之间的平衡。Mustafa 等人^[36]采用了一种基于非单调自适应阈值接受策略的禁忌搜索算法，阈值接受算法（Threshold Acceptance Algorithm, TA）是类似 SA 算法的局部搜索策略，不像 SA 算法那样需要多个参数控制

算法迭代进程,但是仍然需要考虑阈值的合理设定。Kim 等人^[37]提出了一种基于不同参考点的最邻近扫描算法来构造一个质量良好的初始解,之后再利用多种邻域算子从初始解开始进行探索,并且使用类似 LNS 的扰动算子跳出局部最优,实验结果表明了该算法的高效性,并且分析了初始解生成算法对实验结果的影响。Mustafa 等人^[36]设计了一种包含阈值调整和扰动算子的自适应局部搜索算法,该算法由 SA 算法和 VND 算法组成。由于 VRPMPD 问题是 VRPSPD 问题的一种特例,作者在这两个问题的数据集上分别进行了实验,验证了该算法在同时取送货问题上的普适性。Kalayci 等人^[38]设计了一种混合启发式算法解决 VRPSPD 问题,利用蚁群系统的存储结构配合变邻域搜索算法 (Variable Neighborhood Algorithm, VNS) 协同工作, VNS 算法代替蚂蚁释放信息素,而蚂蚁利用信息素为算法提供扰动机制。

逆向物流本身比正向物流提出时间就晚很多,国内关于 VRPSPD 问题的研究最早也在二十一世纪初,多数研究都是由现实生活中的应用场景推动的。衷志远等人^[39]设计了适合取送货一体化配送问题的遗传算法,采用传统的节约法构造了初始解种群。谭巍^[40]建立了 VRPSPD 问题的数学模型,根据问题的特征改造了 2-opt 方法,又结合了蚁群系统形成混合启发式算法,最后实验验证了算法的有效性。曹俊阳等人^[41]在 VRPSPD 问题的基础上,考虑了交通拥堵对车辆速度的影响,建立了这个应用场景下的数学模型,并在 SA 算法中引入局部搜索方法给出了该问题的解决方案。周蓉等人^[42]针对 VRPSPD 问题提出了一种自适应的并行遗传算法,实现深度和广度的同步搜索。盛虎宜等人^[43]研究 VRPSPD 问题在农村电商物流配送上的应用,设计了一种改进的蚁群算法全局规划农村区域内的物流资源。李嘉等人^[44]研究了电动汽车运输的 VRPSPD 问题,分析出载重量和行驶距离是影响电动车能耗的关键因素,将能耗作为成本一起计入目标函数,同时给出基于 SA 算法和 VNS 算法的混合启发式算法作为该问题的求解方案。

1.2.3 VRPSPDTW 问题国内外研究现状

相比于 VRPTW 问题和 VRPSPD 问题,他们的组合问题 VRPSPDTW 受到的关注并不多,目前只有少数几种算法用于解决该问题。Angelesli 等人^[45]是第一个也是目前为止唯一一个用精确算法来解决 VRPSPDTW 问题的学者。他们设计的精确算法主要是经典分支定价法,并且指出 VRPSPDTW 可以看作是 VRPSPD 和 VRPTW 的组合,他们修改了 Solomon 标准数据集^[5],生成了 29 个测试实例并报告了结果,结果没有对比性,并且由于精确算法的局限性,该算法只在 20 个客户点上进行了实验。Lai 等人^[46]提出了一种改进的差分进化算法,并且通过自适应机制调节交叉概率。为了进行试验,他们自定

义了两个测试用例，其中一个只包含 8 个客户点，另一个包含 40 个客户点，本质上还是在小型数据集上进行实验。Boubahri 等人^[47]设计了一种改进蚁群系统，不过他们没有任何实例上进行测试。Lai 等人^[48]提出了一种两阶段混合算法，第一阶段先用改进的蚁群算法专门减少车辆数目，第二阶段用禁忌搜索算法着重减少行驶距离，并且在 40 个客户点的数据集上进行实验，证明了算法的有效性，不过每个阶段的专门化并没有特别指出原因。Wang 和 Chen^[49]正式给出了 VRPSPDTW 问题的混合二进制整数规划模型，并且使用 co-evolution 遗传算法 (coGA) 配合插入算法解决了这个问题，除此之外，他们还提供了 65 个从 Solomon 数据集修改而来的测试实例。实验结果表明，相比于 Cplex 商业软件和基本遗传算法，coGA 能够得到更好的结果，但在 56 个中型规模的实例上，作者并没有将运行时间这一结论展示出来。Wang 等人^[50]使用基于插入的剩余容量和径向附加算法 (Residual Capacity and Radial Surcharge Algorithm, RCRS) 生成初始解，然后用 master-slave 结构实现了并行模拟退火算法，他们的算法在 Wang 和 Chen^[49]的数据集上进行了测试，实验结果与 coGA 相比不仅刷新了多个最优解，最主要的是 CPU 时间显著减少。他们还新增了 30 个大数据集实例，实验结果表明他们的算法能在合理的范围内解决包含高达 1000 个客户的 VRPSPDTW 问题。Liu 等人^[51]针对家庭医疗物流中遇到的车辆调度问题进行了研究，该问题的主体仍是 VRPSPDTW 问题，他们使用遗传算法 (Genetic Algorithm, GA) 和禁忌搜索算法 (TS) 的混合启发式算法解决了该问题，并且在自定义的数据集上进行了实验。最近，J.Hof 等人^[52]使用自适应的大邻域搜索算法 (ALNS) 配合路径重组策略 (Path Relinking, PR) 解决了一系列 VRPSPD 相关的问题，其中也包含 VRPSPDTW 问题，他们在 Wang 和 Chen^[49]的中型数据集和 Wang 等人^[50]的大型数据集实例上分别进行了实验，并得到了非常优越的实验结果。

国内对 VRPSPDTW 问题的研究是从最近 10 年开始的，因此，现有的文献并不多。马庆国等人^[53]使用响应型 TS 算法结合 VNS 算法解决了该问题，实验结果表明，该算法在求解硬时间窗的 VRPSPDTW 问题上是有用的，但实验并没有覆盖所有公开实例。王超等人^[54]设计了 SA 算法和 4 种局部搜索算法解决了该问题，实验表明他们的算法非常高效并可以灵活扩展解决 VRP 问题的其他衍生问题。黄务兰等人^[55]建立了 0-1 混合整数规划模型，并设计了一种全局人工鱼群算法解决了 VRPSPDTW 问题，实验表明该算法比普通人工鱼群算法的效果更好，但是实验没有覆盖全部实例，只挑选了一部分中小型规模的测试实例进行验证。王超等人^[56]利用 2-opt 和路径间搜索法改进了基础布谷鸟算法，该算法在标准数据集上的结果刷新了一部分最优解。徐俊^[57]从 A 运输公司的实际

案例展开研究,对符合该场景的 VRPSPDTW 问题进行建模,并用差分进化算法对其进行求解,结果表明该算法能够在短时间内得到较优的运输方案。张庆华等人^[58]采用引导弹射搜索算法生成初始种群,然后用模因算法进行优化求解,但实验仅仅随机挑选了 10 个测试用例。张思等人^[59]考虑碳排放、能耗和货损等多项成本的 VRPSPDTW 问题,建立了符合该场景的数学模型,用改进的 TS 算法求解了该问题,并验证了改进策略对算法性能的提升。

1.2.4 国内外研究现状分析

通过上述对 VRPSPDTW 问题及其相关问题的研究现状综述,可以总结出以下结论:

1) 由于现实生活中的很多应用场景都离不开时间约束,学术界对 VRPTW 问题的研究热度居高不下。同时,因为逆向物流的概念也是随着社会和环境的发展提出的,所以学术界对 VRPSPD 问题的研究起步较晚,但也产出了大量的研究成果。而相比于 VRPTW 问题和 VRPSPD 问题,两者的联合问题 VRPSPDTW 目前受到的关注并不多,整个研究历史也只有十余载。

2) 由于该问题 Np-Hard 的属性,精确算法只局限于求解小规模问题,一旦客户数目增长,精确算法的计算量将指数上升。而传统启发式算法通常针对问题的特征来构造搜索路径,属于问题依赖型算法,不具有通用性。因此,对于 VRPSPDTW 问题的研究重心放在智能启发式算法上。

3) 国内关于 VRPSPDTW 问题的研究大多都是将其与现实中的某一场景联系起来,解决的是符合该应用场景下的物流配送问题,因此没有可比较的对象。研究 VRPSPDTW 问题本身的算法比较少,而在少有的文献中,实验部分往往只从公开数据集中随机挑选了 10-20 个测试用例,在一定程度上缺乏说服力。

4) 国外关于 VRPSPDTW 问题的严格整数规划模型和标准测试用例都是由 Wang 和 Chen^[49]在 2012 年公开的,此后的研究中,针对该问题的解决方案可以分为两类,一类是基于种群的智能启发式算法^[46,49],在搜索的广泛程度上较好,但最大的局限性就是计算量大,耗时长,因此在现有研究成果中,基于种群的算法解决的都是中小规模的问题。另一类是基于单解的智能启发式算法^[50,52],由于算法是从一个初始解出发,因此算法对初始解的质量要求较高,而且在邻域结构的设计上必须相对复杂,才能使得搜索空间更加广泛。

5) 在现有成果中,无论是基于种群的智能启发式算法还是基于单解的智能启发式算法,都面临搜索广度和搜索深度上的平衡问题,而这一问题仍未解决。

由此可见, VRPSPDTW 问题还有很大的研究潜力, 融合多种约束条件也让该问题更复杂更具有挑战性。

1.3 本文主要创新点

本文通过对 VRPSPDTW 问题研究现状的总结和分析, 针对目前现有算法的不足和局限性, 主要研究高质量解决 VRPSPDTW 问题的智能启发式算法, 具体研究内容和创新点如下:

1) 研究 VRPSPDTW 问题及其智能启发式算法

VRPSPDTW 问题的应用场景十分广泛, 并且降低该配送问题的成本是一个极具挑战性的课题, 本文为 VRPSPDTW 问题构建了严格的数学模型, 并研究了新兴的智能启发式算法。为了高效解决该问题, 本文首先改进了传统扫描算法生成一个高质量的初始解, 让初始解中的路径分布更有可能形成地理位置相对聚集的簇, 为后续算法提供一个良好的开端。接着选择延迟爬山算法作为智能搜索的主导算法, 设计了一种扰动机制, 加强了该算法在后期的搜索能力, 防止其过早收敛。除此之外, 本文设计了多种不同结构的邻域算子, 为探索广泛的解空间进一步提供了保障。

2) 研究搜索广度和搜索深度之间的平衡方法

现有的成果中仍未解决搜索广度和深度之间的平衡问题, 因此, 本文研究了局部搜索策略在搜索广度和搜索深度之间的平衡方法, 引入了强化学习领域中的多臂老虎机智能选择算法来获得搜索过程中的最大收益, 与延迟爬山算法配合进行高效搜索。

3) 将本文针对 VRPSPDTW 问题研究的算法应用于 A 物流公司的物流平台上

为了验证本文算法的有效性和对于现实物流配送的指导意义, 本文收集了 A 物流公司的实际配送需求和实际运输数据, 将本文的算法成功运用在真实物流场景下。

1.4 本文组织结构

本文将围绕 VRPSPDTW 问题的算法研究展开, 一共包含 5 个章节, 具体结构和内容如下:

第一章介绍了本文研究课题的研究背景和研究意义, 调研了 VRPTW、VRPSD 和 VRPSPDTW 三个问题的国内外研究现状, 并且总结了现有研究进展以及存在的问题。最后给出本文在研究内容上的主要创新工作和论文整体组织结构。

第二章描述了 VRPSPDTW 问题的相关理论和针对该类问题的求解算法。由于 VRPSPDTW 问题比传统 VRP 问题的约束条件更多更复杂, 因此在建立其数学模型之

前,首先对该问题的概念进行了简要描述,然后详细介绍了构成该问题的主要元素和特点,接着基于不同要素的侧重点介绍了 VRPSPDTW 问题的多种类别,最后重点介绍了现有的针对 VRPSPDTW 系列问题的求解算法,并总结和分析了每种求解算法的优越性和局限性。

第三章进一步针对 VRPSPDTW 问题进行详细描述,给出了基于该问题的相关假设,介绍了该问题的相关约束条件,接着用数学符号定义了必须的变量与参数,建立了 VRPSPDTW 问题的数学模型,明确了本文的优化目标,为下一章的算法设计奠定了理论基础。最后,结合了现有研究现状中的不足,给出了本文针对该问题求解算法的研究。

第四章详细阐述了本文针对 VRPSPDTW 问题所研究的求解方案,首先概述了整体求解方案的流程框架,然后重点描述了每个求解步骤的核心原理,包括用于生成高质量初始解的改进型扫描算法、用于主导局部搜索的加强型延迟爬山算法、用于平衡搜索广度和深度的多臂老虎机智能选择算法,以及本文设计的多种用于探索解空间的邻域算子。

第五章针对本文提出的求解方案进行仿真实验和实际应用。首先将算法运用在 VRPSPDTW 问题的国际标准数据集上,并将实验结果与现有的算法进行了对比,验证本文提出算法在理论上的可行性和高效性,除此之外,还分析了本文改进的组件对整体算法的影响。接着将本文提出的求解方案应用在 A 物流公司的实际案例上,验证求解方案在现实场景下的可行性。

总结与展望,总结和分析了本文的所有研究工作,并基于其中的不足之处,对未来的研究方向进行了展望。

1.5 本章小结

本章首先介绍了选题的研究背景和意义,强调了取送货一体的物流配送模式广泛的应用场景,以及降低物流运输成本对物流领域的重要意义。然后调研并总结了国内外针对 VRPTW、VRPSD 和两者联合的 VRPSPDTW 问题的主流研究成果,指出了目前对 VRPSPDTW 问题研究的缺乏和不足,最后概述了本文的主要创新点和本文在章节上的内容安排。

第二章 VRPSPDTW 问题的相关理论和求解算法

2.1 引言

由于 VRPSPDTW 问题是一种复杂的 Np-Hard 问题,本章将针对该问题的必要理论知识进行较为详细的描述,首先介绍 VRPSPDTW 问题的定义,紧接着描述该问题涉及到的元素,以及元素的特征或者说属性,然后根据这些元素属性的不同侧重点对该问题进行分类,最后概述了学术界针对 VRPSPDTW 问题提出的求解算法种类,并总结了这些算法的优越性和局限性。

2.2 VRPSPDTW 问题的相关理论

2.2.1 VRPSPDTW 问题的定义

VRPSPDTW 问题是经典组合优化问题 VRP 的衍生问题,VRP 问题描述的是这样一个场景:给定一个车场作为配送中心,一组车辆作为运输的交通工具,为分布在不同地理位置的客户配送(或取回)货物,如何调度这组车辆,使得车辆从车场出发,按一定的次序服务完所有客户回到车场后,整个活动的配送代价(车辆数目、行驶距离或者油耗等成本)最小。并且,车辆在运输的过程中不得超载,每个客户也只能被一辆车服务且服务一次。

VRPSPDTW 问题在 VRP 问题的基础上多加了两个重要约束,第一个约束是时间窗约束,也就是客户预定车辆服务的时间范围,时间窗的起始时间点代表客户希望车辆进行服务的最早开始时间,而时间窗的结束时间点代表最晚开始服务时间,车辆可以在这个时间段内的任意时刻开始服务。本文讨论的时间窗是硬时间窗,即车辆如果比最早开始服务时间更早到,则需要等待,但绝对不允许车辆比最晚开始服务时间更晚到达客户点。第二个约束是取送货约束,在 VRPSPDTW 中,车辆的服务类型不再是单一的取货或者送货,而是两者的混合,并且取货和送货是在同一个客户点发生的,不存在先后关系,这意味着车辆在运输过程中的载重是具有波动性的,而不是单一配送模式下的递增或者递减。

2.2.2 VRPSPDTW 问题的参与元素

在上述对 VRPSPDTW 问题的定义中,可以发现 VRPSPDTW 问题的构成元素主要包含路网、车场、车辆、客户点、约束条件和优化目标。

- 路网

路网即道路网络,也就是配送车辆行驶的道路和环境。在数学模型的抽象下,通常

用一个赋权图来表示整个路网，图中的节点通常表示车场或者客户点，而弧将点与点之间连接起来，构成一条条运输道路，弧上的权重表示两个点之间的运输代价，这个代价可以是单位距离的运输成本、行驶距离或其他衡量指标。赋权图还有有向图和无向图之分，有向图中的有向弧代表真实路网中的单行线，而对应的，无向弧则代表双行线，并且两个方向的弧上权重可以不一样。

● 车场

车场也可以叫配送中心，是一个仓库的概念，管理所有运输车辆的调度，也存储大量货物，是车辆配送的起始点。如果车辆在完成任何后需要返回车场，那么它也是车辆配送的终点。在 VRP 相关问题中，车场可以有多个，每个车场负责管理一部分车辆，从不同车场出发的运输车辆又可以视为不同的 VRP 问题。

● 车辆

车辆是物流配送中装卸货物，完成客户需求的交通工具，一般来说，一个运输车辆具有以下几个属性或者说构成元素：

- 1) 载重。每辆车不可能无限制地装载，因此，车辆具有最大载重上限，在运输过程中不允许超过这个上限。
- 2) 容量。与载重属性相似，车厢的容积是有限制的。
- 3) 类型。车场维护的车辆可以有多种多样的类型，它们在最大载重、最大容量、最长运输距离以及油耗等方面都不尽相同，比如现实物流企业中，最常用的有小面包车、金杯车、大卡车和厢车等。
- 4) 司机。运输车辆是由司机负责驾驶操控的，司机的薪资费用也是物流成本中的一环。
- 5) 成本。一辆车的调度必然会涉及到成本，比如车辆的固定费用、里程费用、燃油费用、司机劳务费用以及车辆维护费用等。

● 客户点

客户点就是 VRPSPDTW 问题中车辆的服务对象的统称，客户点不仅仅指的是个人，还可以是零售商店和分销商等组织，一个客户点通常包括以下属性：

- 1) 位置。客户点的位置通常用经纬度来表示。
- 2) 需求。客户的需求通常只有两种，要么寄送货物，要么接收货物，而货物又有相应的体积和重量等属性。
- 3) 服务时间。车辆在客户点的服务时间取决于客户需求的大小和其他主观因素。

- 4) 时间窗。在 VRPSPDTW 问题中, 客户可以预定一个配送时间范围, 车辆只能在这一时间范围内开始服务。若没有特别说明, 一般都是不可违背的硬时间窗。

● 约束条件

在 VRPSPDTW 问题和相关 VRP 衍生问题中, 车辆运输过程需要满足一系列限制条件, 比如时间窗和车辆载重限制都属于约束条件。根据车辆、客户、路网和其他属性的表现又能产生各种各样的约束, 越复杂的场景下约束条件越多, 同时问题也会更难求解。

● 优化目标

通常来说, 物流配送的优化目标可以围绕物流成本和服务质量展开, 包含以下几个方面或者某几个方面的组合:

- 1) 最小化使用的车辆数目
- 2) 最小化总行驶距离
- 3) 最小化客户的等待时间
- 4) 最小化车辆的空闲时间

2.2.3 VRPSPDTW 问题的分类

根据上述 VRPSPDTW 问题构成元素的不同侧重点, VRPSPDTW 问题又可以分为多类, 如下表所示:

表 2-1 VRPSPDTW 问题的划分类别

构成要素	侧重点	分类
路网	实时变化	静态路网
		动态路网
	道路方向	单行道路网
		双行道路网
车场	数量	单车场问题
		多车场问题
车辆	类型	同构车辆
		异构车辆
	数量	无限车辆
		有限车辆
	是否返回车场	闭合路线
		开放路线
客户	优先级	同优先级
		不同优先级
	需求	只有取或只有送

表 2-1 VRPSPDTW 问题的划分类别（续）

客户	需求	同时有取送货需求
	时间窗	软时间窗
		硬时间窗
	货物	相容货物
		不相容货物
优化目标	目标数目	单目标问题
		多目标问题

2.3 VRPSPDTW 问题的求解算法及分析

VRPSPDTW 问题是一个组合优化 Np-Hard 问题，因此，暴力解法在该问题上是不可行的，从上述国内外研究现状中可以发现，针对 VRPSPDTW 问题的求解方法可以分为四类，即精确算法（Exact Algorithm），基于路径构建的传统启发式算法（Classical Heuristics），智能启发式算法（Metaheuristics）和混合启发式算法（Hybrid Heuristics），其中混合启发式算法是两个及多个传统启发式方法或者智能启发式方法的混合。

2.3.1 精确算法

既然 VRPSPDTW 问题是一个可以构建数学模型的最优化问题，那么该问题就有且只有一个最优解，所谓精确算法就是能够求解该问题精确解的方法。这类方法通常根据数学规划的原理来分析出问题各个元素之间的数量关系，针对 VRPSPDTW 问题的精确算法主要包括分分支定界法、分支切割法、分支定价法、列生成法、割平面法和动态规划法等。精确算法在求解 Np-Hard 问题上的最大局限性就是很难拓展，因为一旦客户数量增加，问题规模变大，精确算法的计算量会指数增长，无法在一个合理的时间范围内得到结果。因此，主流研究的重心迁移到启发式算法这类近似求解算法上。

2.3.2 传统启发式算法

启发式算法的目标是在合理的时间范围内得到令人满意的近似解，离精确解有一定的可接受的偏差。传统启发式算法一般基于路径构造，针对具体问题的特征在一定步数下构造可行解，一般情况下适用于初始解的构造。针对 VRPSPDTW 问题的最常见传统启发式算法有以下有几种：

1) 插入算法

顾名思义，插入算法（Insert Algorithm）以某种规则将客户需求依次插入到车辆中。首先构造一条空路径，即不包含客户，起点和终点都是车场的路径。然后选取一个客户

点作为插入过程的起始，再在不违背所有约束条件的前提下，将剩余的客户点依次插入到该路径内，若该路径对应的车辆无法装载某个客户的需求，那么就再以这个客户点作为下一条空路径的第一个客户点，重复上述的插入过程，直到所有客户点都被安排到路径中。在选择下一个被插入的客户点时，可以随机选择，也可以是根据自定义的规则来选择。

2) 扫描算法

扫描算法 (Sweep Algorithm)^[60] 是一种区域性规划的启发式算法，也属于 “cluster-first-route-second” 方法。它基于客户点的地理位置分布，以车场为参考中心，以水平射线或者与水平线有一定夹角的射线为扫描线，逆时针或者顺时针扫描一圈，将扫描路径内的客户点需求依次装载进车辆中，当一辆车装满或者插入某个客户需求时违反了约束条件，就再开辟一辆新车继续装载，直到所有客户点随着扫描过程的结束全部安排到车辆中。

3) 节约算法

节约算法 (Saving Algorithm) 也叫节约里程法^[61]，它首先将所有客户点看成单独的服务目标，为每个客户点构建一条车场到客户点再回车场的初始路径，这些路径作为一个集合。然后计算任意合并两个路径的代价值或者说行驶距离减少的幅度值，再选取代价最小的合并方案进行路径合并，合并后的路径成为新的路径加入集合中再次参与上述计算，直到集合中的所有路径由于约束条件的限制都无法再被合并。

2.3.3 智能启发式算法

智能启发式算法不针对某一特定场景或问题，更具有通用性。智能启发式算法的灵感很多都来源于自然界的随机现象，目前常用于求解 VRPSPDTW 问题的智能启发式算法有基于种群的和基于单解的，基于种群的主要有遗传算法、粒子群算法、蚁群算法等，而基于单解的主要有爬山算法、模拟退火算法、禁忌搜索算法、大邻域搜索算法等。下面我们将简单介绍几种常见的智能启发式算法。

1) 遗传算法

遗传算法 (GA) 是最早的基于种群的智能启发式算法之一，是从达尔文进化论中 “适者生存和，优胜劣汰” 的观点中获得的灵感^[62]，因此它也叫进化算法。通过概率性寻优，将种群中质量较差的个体逐步淘汰，最后留下质量更好的解，也就是最优解。GA 算法将问题的解看成是一个个体，而将解中的路径看成生物中的染色体，将构成路径的客户点看成基因。首先初始化多个个体从而组成一个种群，然后在种群间通过基因重组

进行个体杂交,产生一系列子代,再利用预先定义好的评价标准对种群中的个体进行评判和筛选,淘汰掉质量差的解,剩余的质量好的解再构成一个新的种群重复这种选择-交叉-变异的步骤,使得种群中个体的适应度越来越高,对应的解的精度越来越好。遗传算法具有非常好的全局搜索能力,搜索范围也十分广泛,并且易于实现。

2) 粒子群算法

粒子群算法(Particle Swarm Optimization, PSO)也是一种进化算法^[63],它由鸟群寻找食物的自然现象所启发,鸟群在寻找食物的过程中会不断地改变自身飞行速度和位置,由分散的状态逐渐聚集,最终找到食物。而粒子群算法中的每个粒子就相当于鸟群中的鸟,也对应于 VRPSPDTW 问题中的一个可能的解,每个粒子一个速度来决定飞行的方向和距离,同时还关联了一个适应值。PSO 算法首先初始化一个粒子群,然后进入循环迭代搜索的流程,每一轮循环中,粒子都会依靠两个极值来更新自己的位置和速度,这两个极值分别是当前粒子找到的最优解,也叫个体最优解,还有整个粒子群找到的最优解,也叫全局最优解。逐渐地,粒子会慢慢靠近历史个体最优解,而整个种群也会向着全局最优解逼近,从而找到最优解。PSO 算法是一种高效的进化算法,避免了复杂的交叉变异操作,也具有一定并行潜力。

3) 禁忌搜索算法

禁忌搜索算法(TS)在 VRP 问题及其衍生问题中的应用十分广泛^[64],它的灵感来自于人类的思维过程,从一个初始解开始,在其邻域内寻找更好的解,迭代这个过程直到找到最优解。TS 算法中最关键的是采用了禁忌技术,禁忌技术要求算法不做或者尽量避免重复工作,用一个列表来记保存算法已经搜索过的历史最优解。这些历史最优解也是局部最优解,这个列表叫做禁忌表,顾名思义,禁忌表中的解会在后续的搜索中被禁止,也就是说,在后续的迭代中将不再搜索或者选择性地使用这些候选解,从而使得搜索效率显著提高,也避免了算法过早陷入局部最优的局面。

4) 模拟退火算法

模拟退火算法(SA)的灵感来源于冶金过程中固体退火的原理,是一种基于单解的概率性局部搜索策略^[65]。在冶金过程中,加热使得固体温度升高,其中的原子处于一种混沌无序的状态,当温度冷却下来之后,内能减小,原子的状态趋于稳定,当温度降至常温时,系统内能达到最小值。在 VRPSPDTW 问题中,从随机无序的解空间找到状态稳定的全局最优解的这个过程和退火过程如出一辙,从最高温度出发进行迭代搜索,随着温度的徐徐冷却,随机寻找解空间中的全局最优解。SA 算法的最大特点就是概率突

跳性,即算法可以概率性的跳出局部最优,从而更有可能寻找到下一个更好的局部最优解。SA 算法的求解效率高且实现简单,但初始温度、冷却率和突跳概率都是需要调节配合的,而这些参数的组合将直接影响 SA 算法的能力和得到的最终结果。

5) 大邻域搜索算法

大邻域搜索算法(LNS)由 Shaw 引进到 VRP 问题中^[20],并且在 VRP 相关衍生问题中的运用已经相当广泛。邻域其实是基于单解的智能启发式算法中一个必要的概念,可以理解为当前解的邻居组成的集合,而邻居是在当前解的基础上进行一定结构变化操作后产生的,一般来说,当前解的邻居和当前解只有轻微不同,而大邻域搜索算法的特点就是每次迭代产生的邻域范围比较大,对于当前解的“破坏”和“重组”过程尤为明显,往往都要从当前解中移除一定比例的客户点,然后再以某种策略重新插回去,造成当前解结构的大规模变化。正因如此,比起其他邻域规模较小的智能启发式算法,LNS 算法的耗时更长。还有一部分学者研究了自适应的 LNS 算法(ALNS),自适应地选择每轮迭代中使用的插入算子和移除算子,使得算法的性能得到一定程度的提升。

2.3.4 混合启发式算法

正是由于智能启发式算法在求解 VRPSPDTW 问题以及相关组合优化问题上比其他种类的算法具有通用性的优势,它们很容易被互相组合,形成混合启发式算法(Hybrid Meta-Heuristic)。在本文 1.2 节的国内外研究现状中可以发现,混合启发式算法在 VRP 相关问题上的使用频率非常高,因为两种通用的智能启发式算法的混合,往往能将两种算法的优势结合起来,互相弥补不足,比如 TS 算法和 LNS 算法的混合,既能利用 TS 算法的快速收敛能力,又能利用 LNS 广泛的邻域搜索能力,充分发挥了他们各自的优势,使得整体算法的性能更好。混合启发式算法并不局限于智能启发式算法的混合,在单一智能启发式算法中混入一些局部搜索算法或者扰动机制,也可以称之为混合启发式算法。

2.3.5 求解算法分析

在上面介绍的针对 VRPSPDTW 的求解方法中,精确算法是最早的一批求解方案,但由于该问题的 Np-Hard 特征,当客户规模增大以后,精确算法会面临指数爆炸的问题,很难甚至无法在合理的时间内得到精确解。传统启发式算法都是基于路径构建的解决方案,针对问题的特征以及节点之间的联系,在具体问题的可行集合中以一定步数构造可能的解。但传统启发式算法的输出比较固定,也就是说,只要给定了问题的输入,算法

就会按照固定的步骤继续往下构造路径，反映出来的就是传统启发式算法是“问题依赖型”算法，在 Np-Hard 问题中得出的结果往往与最优结果有较大的偏差，因此，传统启发式算法更适合用于初始解的构造。而智能启发式算法加入了随机因素，不依赖问题的特有特征，所以具有更广泛的应用，智能启发式算法多数来源于自然界的灵感，比如模拟退火算法、遗传算法、粒子群算法、爬山算法等等，虽然目前还没有从数学理论角度证实智能启发式算法能求解出最优解，但他们在 VRP 问题及其衍生问题中的运用已经表明了其高效的搜索能力。

智能启发式算法虽然能够高效地解决很多 Np-Hard 问题，但是每种算法都有其优越性和局限性，比如模拟退火算法虽然易于实现，也能够有效的跳出局部最优，但算法非常依赖参数的设置，而参数的设置又和问题本身或者数据实例相关；而基于种群的遗传算法全局搜索能力很强，但同样面临参数不易调节的问题，同时，遗传后期子代的个体差异性小，算法容易早熟；禁忌搜索算法拥有记忆能力，搜索效率比相较于其他算法来说要高得多，同时它既拥有较好的局部搜索能力，又在全局搜索方面表现优异，但是搜索的精度依赖于初始解和候选解的生成方式。

从目前的研究来看，混合启发式算法能够将两种启发式算法的优势结合起来，来弥补单一智能启发式算法在收敛速度或者搜索能力上的不足。同时，在单一智能启发式算法中混合一些额外的扰动机制也能针对性地加强该算法的能力。因此，混合启发式算法是求解 VRPSPDTW 问题的趋势。

2.4 本章小结

本章是 VRPSPDTW 问题的相关理论描述和求解方法概述，首先简单描述了 VRPSPDTW 问题本身的定义，对定义中的构成元素展开叙述，然后针对构成元素的不同侧重点，对 VRPSPDTW 问题进行了分类。接着概述了求解 VRPSPDTW 问题常用的几类算法并描述了执行步骤，最后分析了每种算法的优越性和局限性，并指出求解 VRPSPDTW 问题最合适的方式是混合多种启发式算法，或者在单一启发式算法中添加弥补机制。

第三章 VRPSPDTW 问题模型构建及算法研究

3.1 引言

VRPSPDTW 问题是 VRPTW 问题和 VRPSD 问题的组合，同时具有两个问题的特征，反映到数学建模上就是 VRPSPDTW 问题融合了时间窗的约束条件以及取送货一体化的约束条件。图 3-1 清晰地给出了 VRPSPDTW 问题的服务路径（省略了车场），运输车辆从车场出发后依次去客户点服务，客户既有需要取的包裹或者物件，又有需要寄出的物品，并且每个客户有一个预约上门服务的时间窗，车辆运输过程中必须同时满足时间窗约束和车辆最大载重的约束，否则路线不可行。图 3-1 的图（b）展示的路线就违背了时间窗约束，因为车辆服务完客户 2 后的时间显然无法再去客户 3 处服务。（c）图则是违背了载重约束，车辆由于要配送货物，原本就有一个初始载重，而客户 1 的回收物品重量就已经达到了车辆最大载重，因此该路线是不可行的。

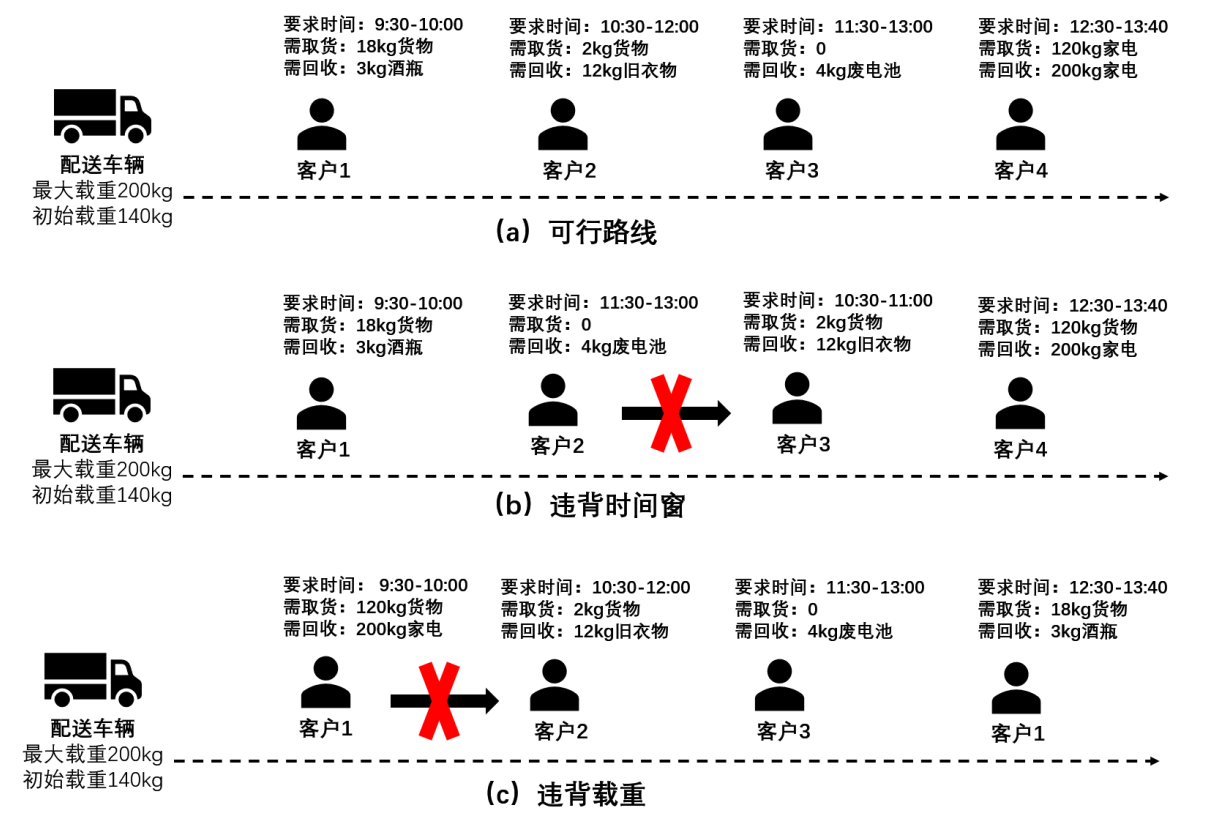


图 3-1 VRPSPDTW 问题路线示意图

本章针对 VRPSPDTW 问题的基本特征，为其构建数学优化模型，指出问题的所有约束条件和优化目标，为后续求解方案的设计奠定理论基础。

3.2 基本假设

本文在第二章的 2.2 节中清晰地给出了 VRPSPDTW 问题的定义, 即 VRPSPDTW 是一个 Np-Hard 的最优化问题, 运输车辆以车场为起点和终点向一组客户提供服务, 在满足一系列前提和约束条件的情况下, 如何安排车辆的行驶路线从而使得这次配送活动的代价最小。根据车场、客户、路网、车辆等属性的不同侧重点, VRPSPDTW 问题具有多种分类, 而本文所研究的 VRPSPDTW 问题是基于以下假设的:

- 1) 道路是静态且对称的, 即两个节点之间可双向行驶并且行驶距离和成本一致;
- 2) 运输车辆不考虑货物的装载方式, 即车辆的装载限制只考虑最大载重量;
- 3) 车场有且只有一个;
- 4) 每个客户的优先级相同;
- 5) 客户数量和客户的需求在配送之前已经确定;
- 6) 客户规定的时间窗属于硬时间窗, 即车辆允许等待但不允许迟到;
- 7) 所有客户都必须被服务;
- 8) 每个客户只能被一辆车服务一次, 不存在重复服务的情况;
- 9) 运输车辆的所有规格都是一致的;
- 10) 运输车辆的数目不做限制;
- 11) 运输车辆的运输速度是匀速的;
- 12) 车辆从车场出发, 且最后返回车场;
- 13) 车辆运输过程中的最大载重不可违背;
- 14) 车辆在客户点取送货时, 先卸货再装载货物;
- 15) 优化的第一目标是最小化配送车辆的数目, 第二目标是所有车辆的整体行驶距离。

3.3 变量的数学表示

本文研究 VRPSPDTW 问题所涉及的所有变量都将用如下数学符号表示:

图:

$G(N,A)$: 一个无向赋权图, 表示路网。

集合:

$N=\{0,1,2,\dots,n\}$: 图中的节点集合, 0 表示车场, 其余编号的节点表示客户;

$A=\{(i,j), i,j \in N; i \neq j\}$: 图中的弧集合, 连接两个节点, 表示客户之间或者客户和

车场之间的路线;

$C=\{1,2,\dots,n\}$: 客户点集合, $N=\{0\}\cup C$;

$V=\{1,2,\dots,m\}$: 运输车辆集合。

参数:

n : 客户数量;

m : 车辆数量, 可视为无限;

Q_k : 车辆 k 的最大载重量;

d_{ij} : 客户点 i 和客户点 j 之间的行驶距离, $i, j \in N, i \neq j$;

t_{ij} : 客户点 i 和客户点 j 之间的行驶时间, $i, j \in N, i \neq j$;

τ_{ik} : 车辆 k 在客户点 i 处的实际服务时刻, $i \in C$;

s_{ik} : 车辆 k 在客户点 i 处的服务时间, $i \in C$;

a_i : 客户点 i 预定的时间窗起始时间, 代表的是车辆的最早开始服务时间, $i \in C$;

b_i : 客户点 i 预定的时间窗结束时间, 代表的是车辆的最晚开始服务时间, $i \in C$;

p_i : 车辆在客户点 i 处的取货需求量, $i \in C$;

d_i : 车辆在客户点 i 处的送货需求量, $i \in C$ 。

变量:

x_{ijk} : 决策变量, 若车辆 k 经过了客户点 i 和客户点 j 之间的弧, 则 $x_{ijk} = 1$, 否则

$$x_{ijk} = 0, \quad k \in V, i, j \in N, i \neq j。$$

Q_{0k} : 车辆 k 离开车场时的初始载重;

Q_{ik} : 车辆 k 在客户点 i 处取送货之后的载重。

3.4 VRPSPDTW 问题的数学模型

结合文献[5]的 VRPTW 问题的数学模型和文献[49]中给出的 VRSPD 问题的数学模型, 本文给出的 VRPSPDTW 问题的数学模型如下:

优化目标:

$$\text{minimize} \quad \sum_{j=1}^n \sum_{k=1}^m x_{0jk} \quad (3-1)$$

$$\text{minimize} \quad \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^m d_{ij} x_{ijk} \quad (3-2)$$

约束条件:

$$\sum_{i=0}^n \sum_{k=1}^m x_{ijk} = 1, \forall j \in C \quad (3-3)$$

$$\sum_{i=0}^n x_{0ik} = \sum_{j=0}^n x_{j0k}, \forall k \in V \quad (3-4)$$

$$Q_{0k} = \sum_{i=0}^n \sum_{j=1}^n d_i x_{ijk}, \forall k \in V \quad (3-5)$$

$$Q_{jk} = (Q_{ik} - d_j + p_j) x_{ijk}, \forall i \in C, \forall k \in V \quad (3-6)$$

$$Q_{0k} \leq Q_k, \forall k \in V \quad (3-7)$$

$$Q_{ik} \leq Q_k, \forall i \in C, \forall k \in V \quad (3-8)$$

$$a_i \leq r_{ik} \leq b_i, \forall i \in C, \forall k \in V \quad (3-9)$$

$$r_{ik} + s_{ik} + t_{ij} \leq r_{jk}, \forall i, j \in C, \forall k \in V \quad (3-10)$$

$$x_{ijk} \in \{0,1\}, \forall i, j \in N, \forall k \in V \quad (3-11)$$

本文研究的 VRPSPDTW 问题的优化目标有两个，分别是公式(3-1)代表的最小化车辆数目，以及公式(3-2)代表的最小化整体行驶距离。最小化车辆数目是第一目标，减少配送分派的车辆数目对物流领域具有重要的意义，因为车辆往往涉及一系列成本，包括车辆的购置费用、折旧费用、保养维修费用、司机费用、燃油费用、碳税等等。最小化整体配送距离是第二优化目标，配送距离也是影响物流成本的直接因素；公式(3-3)保证每个客户只能由一辆车访问一次；公式(3-4)保证了车辆从车场出发，服务完毕之后会返回车场；公式(3-5)确保车辆离开车场时有一个初始负重，这个重量等于该车辆服务的所有客户的送货需求；公式(3-6)表示车辆在配送过程中的载重变化，离开一个客户点时的载重是离开上一客户点的载重减去当前客户点的收货需求再加上寄货需求；公式(3-7)保证车辆在车场处装货时不会超出车辆自身的最大载重；公式(3-8)保证了车辆在运输过程中的动态载重量不会超过自身最大载重；公式(3-9)确保车辆在客户指定的时间窗内开始服务；公式(3-10)表示了车辆在前后客户点间服务的时间关系；公式(3-11)是决策变量的取值范围。

除此之外，本文给出以下规则来比较两个 VRPSPDTW 问题的解，其中 S_i 和 S_j 表示两个不同的解， i 和 j 都是正整数：

- 1) 如果 S_i 所使用的车辆数目比 S_j 更少，那么无论行驶距离的大小关系如何，都认为 S_i 都比 S_j 更优；
- 2) 如果 S_i 和 S_j 所使用的车辆数一样，但 S_i 的整体行驶距离比 S_j 更小，那么认为 S_i 比 S_j 更优；
- 3) 如果 S_i 和 S_j 在车辆数目和整体行驶距离上都相等，那么认为 S_i 和 S_j 一样优。

3.5 VRPSPDTW 问题求解算法研究

从第一章的背景和研究现状可以看出, VRPSPDTW 问题是经典 VRP 问题的延伸, 也是 VRPTW 问题和 VRPSPD 问题的组合, 相比于这些基础问题, VRPSPDTW 问题更加复杂。第二章中描述了求解 VRP 系列问题的几类常用算法, 并总结了各个常见算法的优越性和局限性, 其中, 精确算法无法扩展到中大规模的问题上; 传统启发式算法依赖问题特征来构造解, 不具有通用性; 单一智能启发式算法存在早熟、参数难以调节等问题。因此, 混合几种启发式算法或者在单一智能启发式算法中混合弥补机制是目前最好的选择。鉴于基于种群的智能启发式算法相比基于单解的智能启发式算法更“重”, 计算量更加庞大, 运行时间也更长, 因此, 本文主要将重心放在基于单解的智能启发式算法研究上, 整体求解思路如图 3-2 所示, 客户需求和车辆信息将作为算法的输入, 首先生成一个初始解, 再从初始解出发, 利用混合启发式算法进行迭代寻优, 接着针对优化后的解进行一些修复性的后续工作, 最后输出最终结果。

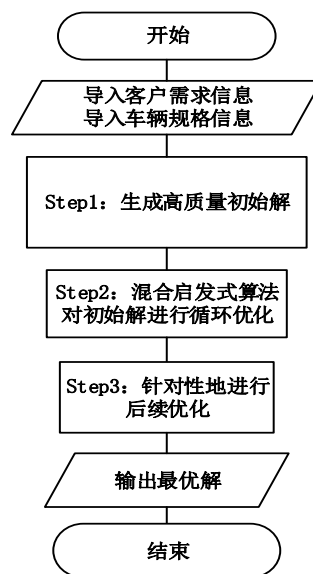


图 3-2 VRPSPDTW 问题求解思路图

基于单解的智能启发式算法都是从一个初始解出发的, 那么如何生成一个高质量的初始解是本文的首要研究内容。从第二章中的相关理论分析可以得知, 初始解的构造往往基于传统启发式算法, 它针对具体问题的特征或者具体数据的分布, 在一定步数下按照某种逻辑来构建可行解。因此, 传统启发式算法属于“问题依赖型”算法。本文在多个经典构造算法的基础上进行了尝试性实验, 发现扫描算法在 VRPSPDTW 问题的初始解构造方面表现得比其他构造算法更好, 于是, 本文选择扫描算法作为生成 VRPSPDTW 问题初始解的方案。在充分了解扫描算法的核心原理后, 本文又继续探究了扫描算法的

改进空间,发现单因素扫描算法的不足,并研究了一种双因素扫描算法来生成更高质量的初始解,以期为整体算法提供一个良好的开端。

生成初始解之后,下一步就正式进入启发式算法的迭代寻优的过程。大多数基于单解的智能启发式算法都具有参数依赖性这一特征,比如模拟退火算法^[70]、阈值接受算法^[71]以及大洪水算法^[72]等等,它们都是在迭代搜索的过程中,以某种策略接受或者拒绝新产生的候选解,直到算法达到停止条件。这些算法的共同特点是,算法搜索的进程强依赖于某个或者某些控制参数,在模拟退火算法中这个参数叫“初始温度”和“冷却率”,在阈值接受算法中这个参数叫“阈值”,在大洪水算法中这个参数叫“水平面”,这种难以调节的控制参数比较依赖问题本身或者实例本身。因此,本文在优化算法方面的研究针对新兴的延迟爬山算法展开,延迟爬山算法是 2017 年由 E.K Burke^[67]提出的局部搜索策略,同时也是一种新兴的智能启发式算法,它的搜索过程只由历史解决定,而不受某些参数控制,这样就避免了难以调节的控制参数对搜索过程的影响,也使得整体算法相对稳定。然而,延迟爬山算法也具有单一智能启发式算法的共同缺陷,那就是容易陷入局部最优,从而可能导致算法过早收敛。通过研究延迟爬山算法的原理和特征,本文的改进思路是针对性地混合一些扰动算子或者局部机制,来弥补其在搜索能力上的不足,激活其在迭代后期的性能。

局部搜索的核心是寻找候选解,在生成候选解方面,本文延续了现有研究中的邻域思想,设计不同结构的多种邻域算子,利用其随机性,尽可能广泛地探索整个解空间。但实际上,在搜索的过程中往往会面临搜索广度和搜索深度的困境,所谓搜索广度就是如何让让算法探索更广阔的空间,体现的是疏散性,而搜索深度则是如何利用现有的信息得到更高精度的解,体现的是集中性。这个问题的实质就是如何利用邻域算子进行高效搜索,大部分文献中的做法都是在每轮迭代中对当前解运用所有邻域算子,生成多个不同方向上的邻居解,再选取其中质量最好的候选解作为下一轮迭代的输入,显然这种方法的重点放在搜索广度上,要求邻域算子的结构相对比较简单,主导的局部搜索策略必须有良好的收敛能力,否则算法运行的大部分时间都将花费在复杂的邻域动作上。换言之,搜索范围越广,算法越不容易收敛到一定精度,而搜索精度越深,算法越容易陷入局部最优,因为更好的解还在更广泛的解空间。因此,搜索广度和搜索深度之间的矛盾,也是本文重点关注的问题。在一部分文献中,变邻域下降算法^[76]被用来跟其他局部搜索策略混合,它的核心思想就是邻域切换,从而达到一种“变则通”的目的,但这个算法的最大局限性在于,邻域结构切换的顺序需要提前指定,而切换的最佳顺序无法

推导。针对这一矛盾,本文从强化学习领域的多臂老虎机问题^[78]获得了灵感,多臂老虎机问题是平衡问题的一个重要模型,和本文所研究的 VRPSPDTW 问题的搜索目标一致,都是在有限的时间内最大化收益,从理论上来讲,针对多臂老虎机问题的解决方案可以移植到本文的平衡问题上来。因此,本文在延迟爬山算法的搜索过程中混合了一种基于多臂老虎机问题的智能选择算法,自适应地解决邻域结构切换的问题,让算法既能在合理的时间范围内收敛,又能让结果达到令人满意的精度。

在前人的绝大部分研究中,优化算法输出的结果就是最终的解,但本文为了挖掘运输成本进一步下降的空间,在优化算法的最后研究了一个修复边界点的过程,为游离在路径边缘的客户点寻找更合适的位置,对应于图 3-2 中的最后一个后续优化的步骤。

至此,本文针对求解 VRPSPDTW 问题的算法研究完毕,整个求解方案完善后如图 3-3 所示,具体的设计细节会在第四章进行详细描述。

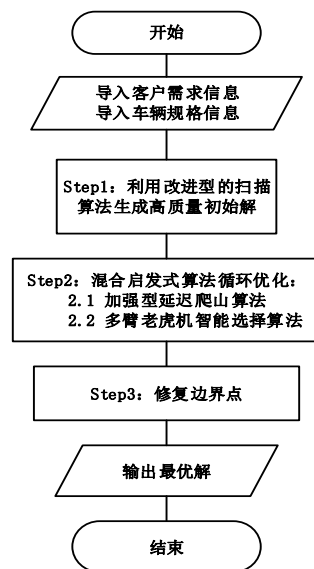


图 3-3 VRPSPDTW 问题求解流程框架

3.6 本章小结

本章主要描述了 VRPSPDTW 问题的数学模型和算法研究,首先概述了 VRPSPDTW 问题的两个重要约束,然后根据第二章中关于该问题属性的不同侧重点,给出本文所研究的 VRPSPDTW 问题的基本假设,接着,总结了 VRPSPDTW 问题涉及到的所有参数和变量的数学符号表示,结合该问题的定义和相关理论,建立了 VRPSPDTW 问题的数学模型,并详细解释了模型中的优化目标、约束条件的含义以及两个解的比较原则。最后,本章针对求解 VRPSPDTW 问题的算法进行了详细研究和分析,并给出了本文的求解方案框架。

第四章 求解 VRPSPDTW 的算法设计

4.1 引言

本文在上一章节中描述了 VRPSPDTW 问题，并且介绍了其数学模型和优化目标。本章将针对上述数学模型，提供一种混合启发式算法，主要由加强型延迟爬山算法（Enhanced Late Acceptance Hill-Climbing Algorithm, ELAHC）和多臂老虎机智能选择算法（MAB）组成。同时，在生成初始解的步骤中，为了消除基本扫描算法中只根据极角扫描造成的不合理性，本文设计了一种具有两个决定性因素的改进扫描生成算法，以生成高质量的初始解。在下面的小节中，首先介绍了求解方案的整体框架，然后详细描述了算法的每个步骤和设计细节。

4.2 求解 VRPSPDTW 的混合启发式算法基本框架

本文提出的混合启发式算法是基于单解出发的，整体可以分为三个步骤：第一步使用改进的扫描生成算法生成一个高质量且可行的初始解，并且在这个过程中保存边界点；第二步是对初始解进行迭代优化，此过程由加强型延迟爬山算法作为主导的局部搜索策略，在多种邻域算子和多臂老虎机智能选择算法的配合下，逐步搜索问题的解空间，直到算法达到设定的终止条件；第三步对第二步优化输出的解进行后续优化，尝试重新安排第一步保存下来的边界点，以期找到成本更低的解决方案。图 4-1 描述了算法的核心步骤。

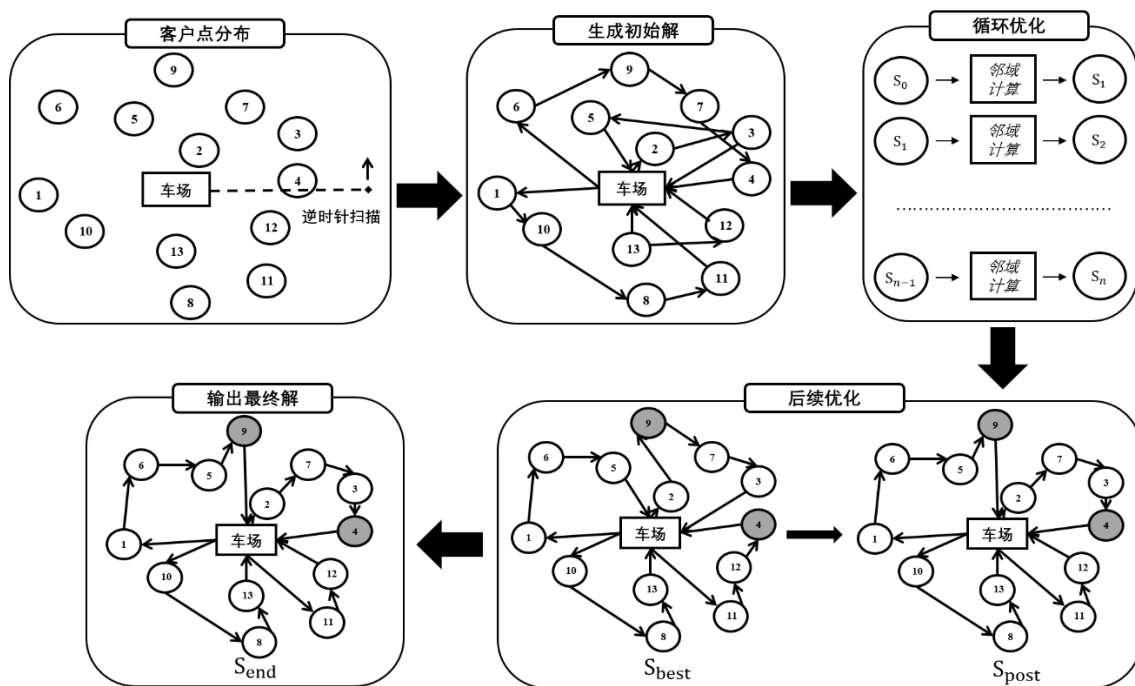


图 4-1 VRPSPDTW 求解算法核心步骤

4.3 初始解生成算法

对于基于单解出发的局部搜索算法来说，一个高质量的初始解能帮助算法跳过一部分无效解或者说差解，从而使得算法能在有限的时间内换取更高精度的最终解。与不确定性较大的随机生成算法相比，本文改进了基础扫描生成算法，该算法能够生成客户点位置相对集中的初始解。基础扫描算法首先由 Gillett 和 Miller^[60]提出。基本思路是以车场为参照点，任意取一个客户点为扫描的起始点并且定其角度为 0 度，也就是从水平线或者任意角度的射线开始，顺时针或逆时针扫描客户点。在所有限制条件都满足的情况下，依次将扫描路径上的客户点插入可用的车辆中，直到所有客户点都安排好，也就是说客户点的插入顺序是按极角来排序的。按照这个逻辑，如果有两个客户点，他们与车场之间的极角相近但实际距离很远，也会被划分到同一路径中，这样显然是不够合理的。如图 4-2 所示，假设车场和客户点的分布如图 (a) 所示，并且扫描算法从水平线开始逆时针执行，那么单独使用极角时，客户点的安排顺序可能如 (b) 所示，但其实客户点 {1,2,3,4} 在距离上更倾向于聚成一簇，如 (c) 所示。

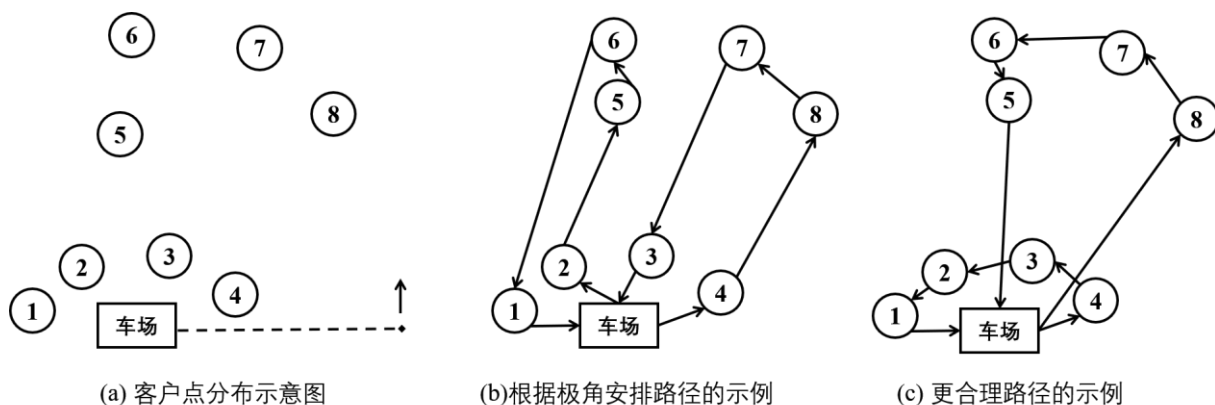


图 4-2 扫描算法示意图

因此，相比于单独使用极角这一个因素作为客户插入顺序的依据条件，我们加入距离这个因素与极角配合使用，控制客户需求插入的顺序。但如果在扫描的过程中，安排一个客户点的需求到车辆之后，下一个考虑装载的是距离当前客户点最近的客户，那么必须在每次插入下一个客户点的需求之前，计算所有剩余客户点和当前客户点之间的距离，才能找到距离当前客户点最近的点，这个过程的复杂度是 $O(n^2)$ ，对于生成初始解来说，计算量是十分庞大的，特别是对于规模较大的数据集来说。所以本文用客户点与车场之间的距离做近似，以车场与任一客户点之间的连线做扫描线逆时针或者顺时针扫描，按照公式(4-1)对需求池中的客户点进行排序，其中， α 和 β 分别表示极角和距离上的权重。从组合值最小的客户点开始，将客户的需求插入可用的车辆中，如果该客户点的

需求插入当前车辆时违背了限制条件，那么就重新开辟一辆可用的空车，从剩下的客户中再次选择与组合值最小的客户点，重复以上步骤，直到所有客户需求都被安排车辆中。这样生成的初始解中，每一辆车上搭载的需求会更大可能构成地理位置相对聚集的簇。

$$x(i) = \alpha \times \text{polarangle}(i) + \beta \times \frac{\text{disToDepot}(i)}{\sum_{i=1}^N \text{disToDepot}(i)} \quad (4-1)$$

4.4 优化算法设计

生成初始解后，下一步是利用局部搜索策略对初始解进行迭代优化，本文中使用的局部搜索策略是由加强型延迟爬山算法(ELAHC)^[67]和多臂老虎机智能选择算法(MAB)^[68]组成的混合启发式算法。其中 ELAHC 算法控制搜索的方向和停止条件，相比于基础 LAHC 算法，ELAHC 设计了一种扰动机制，让算法维持的存储结构中的解更具有多样性，为算法提供了一种跳出局部最优解的重启动方式，从而激活算法在迭代后期的搜索能力。在 ELAHC 算法搜索的过程中，本文设计了多种结构的邻域算子尽可能广泛地探索解空间，但如果毫无方向地探索，会使得算法花费大量时间探索许多无效的解或者差解，可能很难让算法最后收敛到一个令人满意的结果上。因此，为了让搜索广度和深度达到一定程度的平衡，本文使用了从强化学习领域延伸而来的 MAB 智能选择算法，对由不同的动作构成的邻域结构进行自适应交替搜索，根据它们在搜索过程中的表现做决策，以期获得最大收益。最后，本文针对扫描生成初始解的过程设计了一个后续优化过程，目的在于让每辆车中的边界客户点有机会被安排到更合适的车辆中。混合启发式算法的执行流程如伪代码 4-1 所示：

伪代码 4-1 h-LAHC 算法流程

输入： 初始解 S_0

输出： 优化后的解 S_{best}

```

1  初始化  $S_{\text{best}} = S_0$ ,  $S_{\text{current}} = S_0$ ,  $I_{\text{idle}} = 0$ ,  $i = 0$ ,  $L_{\text{list}} = []$ ,  $HB_{\text{list}} = []$ 
2  用  $S_0$  初始化  $L_{\text{list}}$  的所有元素
3  WHILE  $i < I_{\text{max}}$  或者  $I_{\text{idle}} > L_h \times r$  DO
4       $N_i = \text{MABselection}(N_1 \dots N_7)$ 
5       $S_{\text{candidate}} = N_i(S_{\text{current}})$ 
6      IF  $S_{\text{candidate}}$  优于  $S_{\text{current}}$  THEN
7           $I_{\text{idle}} = 0$ 
8      ELSE
9           $I_{\text{idle}}++$ 
10      $j = i \bmod L_h$ 
11     IF  $S_{\text{candidate}}$  比  $L[j]$  更优 或者  $S_{\text{candidate}}$  不比  $S_{\text{current}}$  差 THEN
12          $S_{\text{current}} = S_{\text{candidate}}$ 
    
```

```

13       $L[j] = S_{current}$ 
14  ELSE
15       $S_{current} = S_{current}$ 
16  IF  $S_{current}$  比  $S_{best}$  更优 THEN
17       $S_{best} = S_{current}$ 
18      在  $HB\_list$  末尾追加  $S_{best}$ 
19  IF  $I_{idle} > L_h \times 0.5$  并且 扰动机制未被开启 THEN
20      利用随机的两个邻域算子对  $HB\_list$  中的解进行扰动, 用  $HB\_list$  替换  $L\_list$ 
21       $S_{current} = HB\_list[l \times HB\_list \text{ 的长度}]$ 
22       $I_{idle} = 0$ 
23  END WHILE
24  RETURN  $S_{best}$ 

```

其中 S_0 、 S_{best} 、 $S_{current}$ 、 $S_{candidate}$ 分别表示初始解、最终解、当前解和候选解, I_{max} 表示最大的迭代次数、 I_{idle} 是一个失败计数器, 记录 LAHC 策略连续失败的次数, r 是一个收敛率, 当策略连续失败的次数到达一定数目的时候, 认为算法已经到达瓶颈, 不需要运行完所有迭代次数就可以提前收敛, l 的意义是扰动机制开启时重启动的位置。可以看出整个大循环都是由 ELAHC 算法主导的, MAB 智能选择算法作为一个重要组件嵌入其中, 为每轮迭代提供一个邻域结构。对于每个算法中的细节我们将在下面的小节中详细展开。

4.4.1 加强型延迟爬山算法

1) 基础型延迟爬山算法

延迟接受策略是 Edmund K. Burke^[69]提出的一个新颖的局部搜索策略。它和大名鼎鼎的模拟退火算法^[70]、阈值接受算法^[71]以及大洪水算法^[72]类似, 都是在迭代搜索的过程中, 以某种策略接受或者拒绝新产生的候选解, 直到算法达到停止条件。上述提到的这三种算法的共同特点是, 算法中控制参数的变化都是任意调节的, 在模拟退火算法中这个控制参数叫“初始温度”和“冷却率”, 在阈值接受算法中这个控制参数叫“阈值”, 在大洪水算法中这个参数叫“水平面”, 这种难以调节的控制参数比较依赖于问题本身或者实例本身。而延迟接受策略的搜索过程是由历史解决定的, 而不是由任意定义参数决定的。提出者在后续的研究中完善了延迟爬山算法 (LAHC)^[67], 这个启发式算法是经典爬山算法 (Hill Climbing Algorithm)^[73]的一个延伸, 不同的是, 在经典爬山算法中, 每次迭代生成的候选解总是与当前解进行比较, 如果更优, 就接受它, 并继续向前搜索; 否则直接停止搜索过程。在复杂的问题中, 这样得到的最终结果大概率只是局部最优解, 还有更多搜索空间没有探索到。而在 LAHC 算法中, 新产生的候选解不与当前

解做比较，而是与几代之前的历史解作比较，增加了差解的接受几率，帮助搜索进程跳出局部最优解从而防止其过早收敛。

基础 LAHC 算法的核心思想如图 4-3 所示，它维护一个固定长度为 L_h 的列表 L_list 用于存放 L_h 代以来的历史解。对 L_list 初始化的方式是直接将其中的所有元素初始化为初始解的目标函数值。之后每次迭代产生的候选解将与 L_list 中 L_h 代以前的历史解进行对比，如果更优，则接受它作为下一轮迭代的输入，并且用它替换掉 L_list 中对应的历史解；否则，直接舍弃掉这个候选解。在图 4-4 中，假设 L_h 为 4， L_list 中的 4 个解已经在前 4 轮迭代中被更新成了相应的值，那么第 5 次迭代产生的候选解 S_5 就会和 4 代以前的历史解 S_1 作比较，如果更好，则接受 S_5 ，并用 S_5 的目标函数值替换掉 L_list 中对应的 S_1 的目标函数值。值得注意的是，在 LAHC 策略中， L_list 的更新始终是贪婪的，即只接收更好的解而舍弃更差的解。而且，迭代次数 i 和 L_list 中对应的元素下标 v 之间的对应关系是 $v = i \bmod L_h$ ，因此，第 9 次迭代中产生的候选解 S_9 将与 L_list 中的第一个元素作比较。

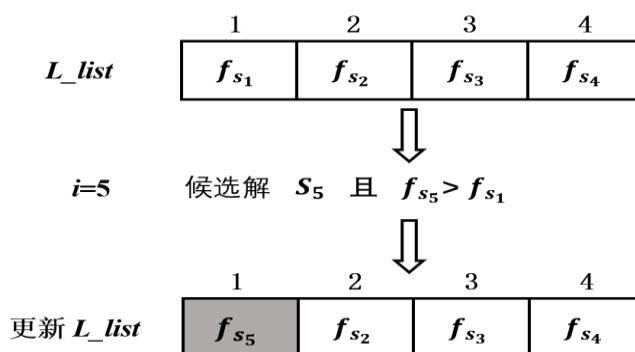


图 4-3 基本 LAHC 说明图

2) 加强型延迟爬山算法

通过上一节对基础 LAHC 算法的描述我们可以知道，LAHC 算法中最重要的组件就是一个维护历史解的短期存储结构 L_list ，在优化过程中，这个数组会不断被更好的解更新，算法最终会朝着局部最优解收敛。那么试想一下，当算法进入迭代后期，这个存储结构中会分布着什么样的解呢？不难想到，随着算法的推进，存储结构中的每个位置最终被更新成局部最优，并且算法会陷入瓶颈，因为存储结构中已有的历史解已经非常优秀，那么新产生的解将大概率会被拒绝，算法也将停止，而这在依靠随机性搜索的智能启发式算法中，这一停止可能意味着算法早熟。所以想要再次激活算法在迭代后期的表现，我们需要这个存储结构中的解最好不具有特别优越的质量并且能具有多样性，值得注意的是，这里所指的“不具有特别优越质量的解”并不代表差解，因为差解不阻碍

候选解的接受。

要实现这个思想,最简单的想法是将其中的解全部更新成随机生成的解,这样 L_list 中的解就极具多样化,但随机生成的解的质量是无法控制的,算法很容易又从一个不可知的方向重新开始。因此,本文的做法是保存算法执行期间产生的所有历史最优解,在算法后期用它们来更新 L_list ,由于历史优解互不相同,正好构成了多样性,而且解的质量既不是极其优秀又不算差,仍然可以让搜索进程沿着原有的方向而不是从头开始。考虑到要经历多次迭代才可能产生一个局部最优解,算法执行过程中的所有局部最优解可能都无法填充整个 L_list ,因此,不够填充的部分本文通过对历史最优解进行扰动形成新解来循环填充,具体方法是选择两个随机的邻域算子(见 4.4.2 节)运用在随机选择的历史优解上,这个过程只需要随机,并不要求新生成的解多么优秀。算法的示意图如图 4-4 所示,当算法进入迭代后期时(图中用 $i = n$ 表示),我们用记录历史最优解的 HB_list 去替换已经陷入瓶颈的 L_list ,然后在这个新的 L_list 的基础上继续进行延迟比较,使算法跳出局部最优的困境,本文将这一算法称作加强型 LAHC 算法(ELAHC)。扰动的步骤在伪代码 4-1 中的 18-22 行。与基础 LAHC 算法不同的是,基础 LAHC 维护的存储结构中存储的是解的目标函数,而 ELAHC 算法为了方便后期做扰动,存储结构中存储的是解本身。值得一提的是,整个加强过程是通过扰动 L_list 让算法获得重新启动的机会,因此,ELAHC 算法无疑会比基础 LAHC 算法执行更多的迭代次数,也会消耗更多的运行时间。

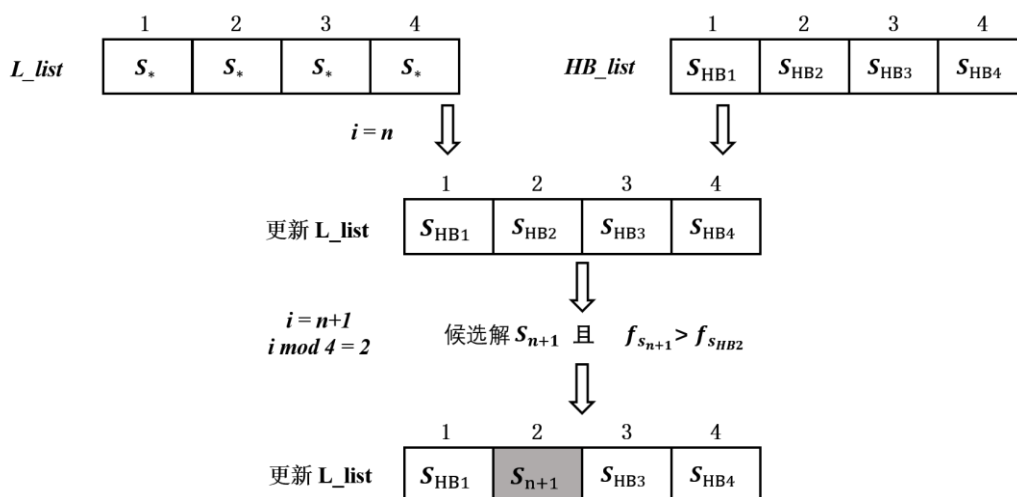


图 4-4 ELAHC 说明图

4.4.2 邻域算子

邻域是智能启发式算法的一个必要的概念，可以理解为当前解的邻居组成的集合，而邻居是在当前解的基础上进行一定变化操作后产生的，而用于产生当前解邻居的操作就是邻域算子。本文参考了 2-opt、shift/swap^[5]算子和减少车辆的策略^[74]针对所研究的 VRPSPDTW 问题设计了七种邻域算子，分别是弹式链算子、路径内重定位算子、路径间重定位算子、路径内交换算子、路径间交换算子、最短路径移除算子和最差路径移除算子。其中前五种都是基本算子，后面两种移除路径的算子属于目标导向算子，专门用来减少配送时的运输车辆数目。它们在一定程度上能随机生成当前解决方案的邻居，引导优化过程探索更广阔的搜索空间。

邻域算子的核心实际上是将路径“拆毁”再“重组”的过程，在描述邻域结构之前，首先介绍用于“拆毁”路径的移除算子和用于“重组”的插入算子。

- 随机插入：在某一车辆或整个解决方案的任意位置随机插入一个客户。
- 贪心插入：在使得某一车辆或整个解决方案内行驶距离增加最少的位置插入一个客户。
- 随机移除：随机移除某辆车或整个解决方案中的一个客户。
- 最坏移除：移除某一车辆或整个解决方案中导致行驶距离增加最多的客户请求。

下文的描述图中，圆圈和其中的数字表示客户点，路径中被深色标记的圆圈表示的是当前邻域算子需要操作的客户。

1) 弹式链算子

弹式链算子^[75]用于重复地将客户请求从一条路径移动到另一条路径，这种重复性操作发生在给定数量的路径（链长）中。它先选择某条配送路线作为弹式链的起始，该配送路线也叫第一条路径，然后弹出（移除）该路径中的某个客户，再将该客户插入到下一条配送路线中，也叫第二条路径，接着从第二条路径中再移除某个客户，又将其插入到下一条配送路线中。重复上述过程，直到从链长中的最后一条路径中弹出的客户插入到第一个路径中，形成一个闭环。以上的选择、插入和移除操作都是随机的。值得注意的是，如果弹出的请求不能插入下一条路径中的任何位置，将会重新随机选择一个请求或者重新选择一条路径，直到插入操作可行。路径的选择和弹出请求的选择都是随机的，而插入方法是贪婪的。图 4-5 为弹式链算子执行的基本过程。假设链长为 3，即整个解中有三条路径参与变换，第一条路径中的客户 {2} 被弹出，并插入到第二条路径中，客户 {8} 从第二条路径中被弹出并插入到第三条路径中，最后一条路径中的客户 {6} 被插入到

第一条路径中，从而生成一个与当前解不一样的邻居解。

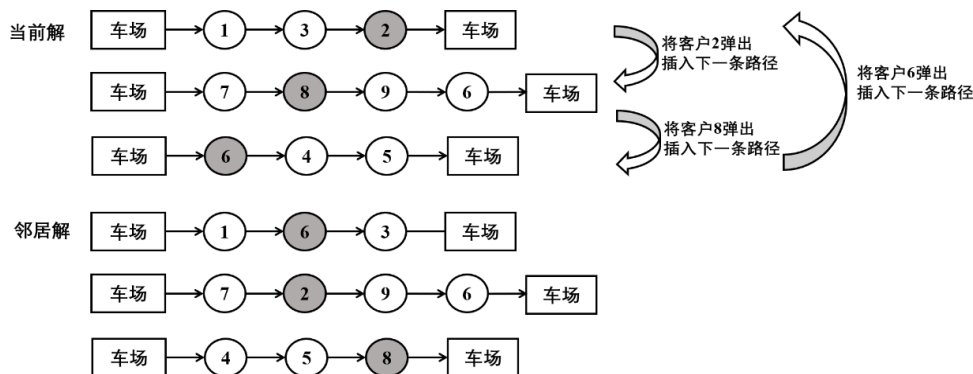


图 4-5 弹式链算子执行过程示意图

2) 路径内重定位算子

路径内重定位算子随机选择整个解决方案的一条路线，并通过最坏移除的方法移除路径上的客户。随后，被移除的客户将通过贪婪的方式插入到完全相同的路径中，简而言之就是将路径内的某一个客户重新安排到一个更合适的位置。在这种情况下，被移除的客户绝对可以重新插回原路径中，因此，不用为该算子考虑插不回去的备用方案。图 4-6 显示了一条路径在执行路径内重定位算子前后的变化情况。客户 {2} 的位置在原路径中发生了变化。

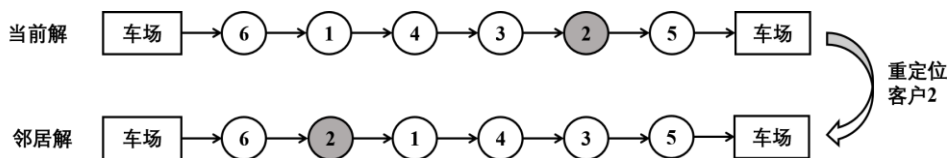


图 4-6 路径内重定位算子执行过程示意图

3) 路径间重定位算子

路径间重定位算子和路径内重定位算子之间的主要区别是，被移除的客户将被插入到整个解的另一个随机选择的路径中，而不是插回原路径中。移除和插入都是随机的。如果被移除的客户无法插入所选路径中的任何位置，则将它随机插入到其他路径中的任何位置。图 4-7 为路径间重定位算子的执行过程，其中第一条路径的客户 {2} 被重新安排到第二条路径中。

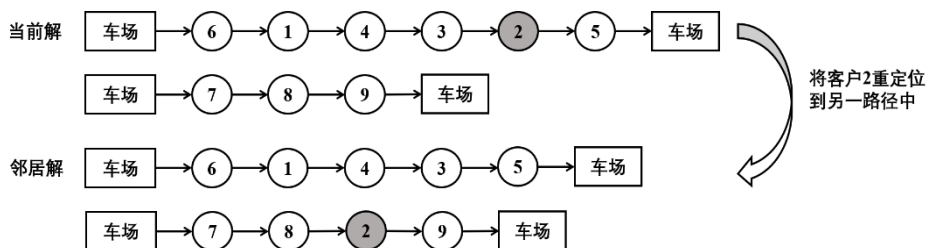


图 4-7 路径间重定位算子执行过程示意图

4) 路径内交换算子

路径内交换算子在同一路径中交换两个客户的位置，路径和两个客户都是随机选择的。但是一旦可行路径中的客户配送顺序确定，简单地交换两个客户的位置很容易使路径变得不可行。因此，这里的“交换”有一个广义的含义，如果产生了一个不可行的路径，将回滚这次交换操作，并将这两个客户随机插入到原路径中的任意位置，相当于连续执行两次路径内重定位算子。图 4-8 展示了路径内交换操作的过程，请求{1}和请求{2}在同一路径中的位置交换过程。

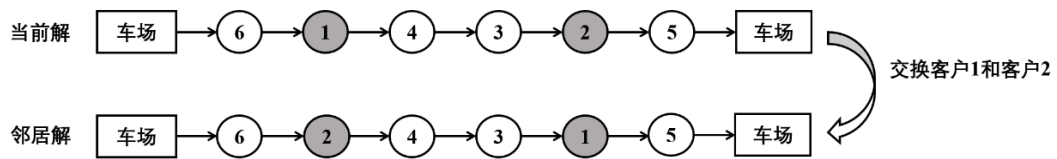


图 4-8 路径内交换算子执行过程示意图

5) 路径间交换算子

路径间交换算子分别从两条路径中选一个客户交换到对方路径中。两条路径的选择和两个客户的移除都是随机的，而插入过程是贪婪的。与路径内交换算子一样，若插入不成功，将考虑不可行情况下的策略。如果在目标路径中找不到可行位置，则将该客户全局随机地插入到整个解中。图 4-9 为路径间交换算子的执行过程。

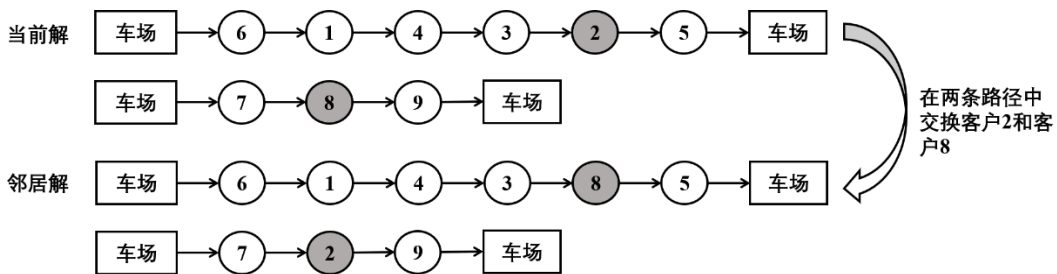


图 4-9 路径间交换算子执行过程示意图

6) 最短路径移除算子

整个解是由一定数量的车辆组成的，而每个车辆对应一条路径，最短路径移除算子的目标是为整个解减少一条路径，也就是减少一辆车的使用。最短路径的含义是服务客户数量最少的路径，将它移除后，该路径中的客户将被打乱顺序并依次重新插回整个解的任意位置。成功的情况是所有的客户都被重新安排进其他路径中，而失败的情况是某个或者某些客户恰好不能被插入到任何路径中，那么将为不能插入的客户重新开辟一辆车。

7) 最差路径移除算子

一般来说，车辆服务的客户越多，行驶的距离就越远。因此当一辆车在运载的需求

数量少, 行驶距离却很大的情况下, 本文认为这辆车的性价比最低, 也就是所谓的“最差”, 这个特性可以用车辆的行驶距离与它携带的客户数量之比来表示, 比率最大的车辆认为是“最差”的车辆, 因此这辆车上的所有客户将会随之移除, 并被插入到整个解中更合适的位置, 插入的具体过程跟最短路径移除算子一样。

这七种不同结构的邻域算子都将以不同的方式产生当前解的邻居, 构成当前解的邻域。前五种邻域算子是基本邻域算子, 主要以破坏并改变路径的结构为主, 其中弹式链算子会较大地改变当前解, 其余算子都是单操作, 一次迭代中也可以让它连续执行多次, 利用随机因素来产生不同邻居解。后面两种邻域算子是目标导向型邻域算子, 目标就是减少使用车辆的数目, 其中最短路径移除算子专门用来减少客户点少的路径, 最短路径中的客户点越少, 效果越好。而最差路径移除算子专门用来发现不合理的路径, 就算可能因为最差路径中的客户点数量比较多, 不能成功将整条路径移除, 也能重新安排部分客户, 减少不合理路径带来的额外运输成本。

4.4.3 智能选择算法

上文中描述的 ELAHC 算法提供了一种稳定收敛的局部搜索策略, 它的规模无关性、问题无关性和控制参数唯一性这几个特征都已经得到验证, 正因如此, 它非常容易并且适合与其他策略进行配合。而本文使用了七种邻域算子来为 ELAHC 算法提供候选解, 有针对路径内的、针对路径间的和针对整个解的。那么下面考虑的问题是, 怎么利用这些邻域算子进行高效的搜索。实际上, 在搜索的过程中往往会面临搜索广度和搜索深度的困境, 搜索广度就是算法如何在当前解的邻域中广泛地搜索, 体现的是疏散性, 而搜索深度则是如何利用现有的信息让算法趋向于得到一个更高质量更高精度的结果, 体现的是集中性。大部分文献中的做法都是在每轮迭代中对当前解运用所有邻域算子, 生成多个不同方向上的邻居解, 再选取其中质量最好的候选解作为下一轮迭代的输入, 显然这种方法重点放在广泛探索上, 要求邻域算子的结构相对比较简单, 主导局部搜索策略必须有良好的收敛能力, 否则算法运行的大部分时间都花费在复杂的邻域动作上。换言之, 搜索范围越广, 算法越不容易收敛到一定精度, 而搜索精度越深, 算法越容易陷入局部最优, 因为更好的解还在更广泛的解空间。因此, 在搜索广度和搜索深度之间的平衡, 也是本文重点关注的问题。下面针对这个问题, 首先介绍运用得十分广泛的变邻域下降算法 (VND), 然后引出本文使用的多臂老虎机 (Multi-Arm Bandit Algorithm, MAB) 智能选择算法。

1) 变邻域下降算法

VND 算法也是一个局部搜索策略^[76], 被广泛应用于 VRP 问题及其衍生问题中, 它利用不同结构的邻域算子进行交替性搜索, 尝试在疏散性和集中性之间达到平衡。邻域算子的变换顺序由它们结构的复杂度决定, 由复杂度最小的邻域算子开始, 如果所找到的解决方案优于目前的解决方案, 则再次从第一个邻域算子开始, 否则, 切换到下一个邻域算子, 直到没有更多的邻域算子可用为止, 图 4-10 表示了 VND 算法的过程, 即“变则通”, 毕竟每个邻域中的局部最优解不一定都相同, 但全局最优解可能出自于某个邻域的局部最优解。

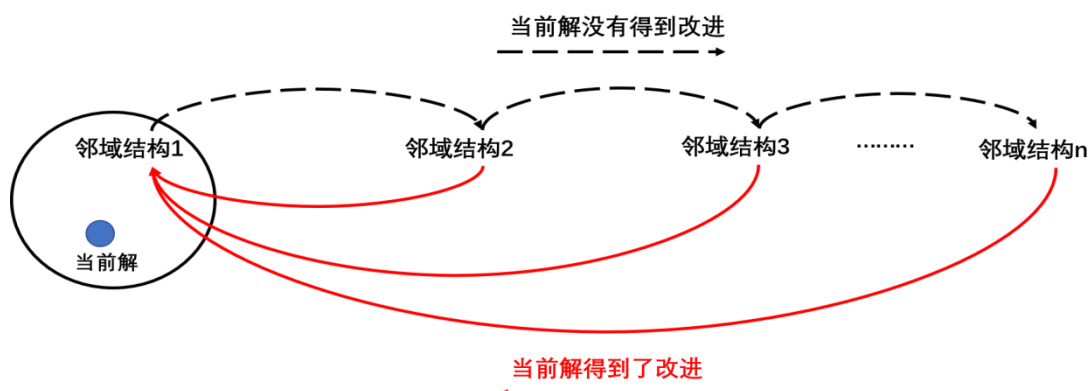


图 4-10 VND 算法说明图

如果把局部搜索比作为爬山, 那么 VND 算法每切换一次邻域就相当于变换了地形, 搜索的过程在不同地形中跳跃。但是, VND 算法的最大局限性在于邻域的变换顺序是事先定好的, 这种顺序不一定总能找到最优解, 因为最佳的邻域变换顺序可能依赖于问题本身或者实例数据本身^[77], 而这些信息对于一个利用随机性和不确定性搜索的局部搜索策略而言, 是非常不容易判断的。

2) 多臂老虎机智能选择算法

本文使用从强化学习领域的多臂老虎机问题中引申而来的 MAB 智能选择方法, 以某种策略来动态的选择运用到每轮迭代中的邻域算子, 在有限的时间或者说有限迭代次数内, 尝试在搜索广度和搜索深度这两个方面上尽量达到平衡, 从而最大化搜索过程中的收益。

多臂老虎机问题^[78]为探索和利用困境 (Exploitation and Exploration Dilemma) 提供了一个清晰的理论模型, 也是概率论中的经典问题, 其解决方案也被广泛用于 AOS (Adaptive Operator Selection, AOS) 和 MOP (Multi-Objective Problem, MOP) 这种决

策问题中^[79-81]。多臂老虎机问题假设一个赌徒在玩多台老虎机，每一个老虎机的摇臂都有一定的回报，问题的目标就是在一定的回合内选择老虎机的臂来使赌徒的利益最大化。在这个问题的多种解决方案中，置信区间上界法（Upper Confidence Bound, UCB）^[82]是最受欢迎的，它的主要思想是综合考虑一个臂上的平均收益以及收益稳定性，而不是将机会均摊给每一个臂（Exploration）或者根据已有的结论每次都选择最好的臂（Exploitation）。我们使用到的 MAB 智能选择算法是 UCB 算法的直接实现，每个邻域算子相当于多臂老虎机问题中的每个摇臂，为每个邻域算子关联一个实证质量，代表该邻域算子从第一次迭代到当前迭代以来的平均表现，这个实证质量将会在每一次迭代中按公式 4-2 更新。

$$q_{i,ns} = \frac{n_{i-1,ns} \times q_{i-1,ns} + r_{i,ns}}{n_{i,ns}} \quad (4-2)$$

其中 $q_{i,ns}$ 代表邻域 ns 在迭代次数 i 时的实证质量， $n_{i,ns}$ 是指该邻域算子截至迭代次数 i 时被使用的次数， $r_{i,ns}$ 表示的是该邻域用于当前迭代时的表现，即优化程度，计算方式如公式 4-3 所示。

$$r_{i,ns} = \begin{cases} \left(1 + \frac{NV_{i-1} - NV_i}{NV_0}\right) \times q_{i-1,ns}, & \text{若车辆数变化} \\ \frac{TD_{i-1} - TD_i}{TD_{i-1}}, & \text{车辆数不变, 行驶距离变化} \end{cases} \quad (4-3)$$

其中 NV_i 表示当前迭代中当前解包含的车辆数目，那么 NV_0 表示初始解包含的车辆数目， TD_i 表示当前迭代中当前解的整体行驶距离。

接下来，根据公式 4-4 来选择每轮迭代使用的邻域算子。这个公式包含两个重要部分，由加号分隔开，一个是邻域的表现，由实证质量 $q_{i,ns}$ 表示，体现的是搜索的深度，另一个是根据该邻域至今被使用的次数给出的置信区间上界，体现的是搜索的广度，当邻域目前为止使用的次数（ $n_{i,ns}$ ）越少，越有机会被选择，从而让搜索过程跳跃到该邻域所在的“地形”上。而 C 是一个常量，用于平衡这两个方面。

$$NS(i) = \max_{ns=1 \dots N} (q_{i,ns} + C \times \sqrt{\frac{2 \times \log \sum_{j=0}^N n_{i,j}}{n_{i,ns}}}) \quad (4-4)$$

4.5 后续优化

后续优化过程主要对应于扫描生成初始解的过程，在用扫描算法生成初始解时，一辆车装下客户 A 的需求后刚好装不下客户 B 的需求（不满足时间窗限制或者车辆载重限制）时，客户 B 的需求需要另起一辆车装载。那么这种情况下，客户 A 和客户 B 在

地理位置上可能相近,但由于扫描算法的规则被装载到了不同车辆中,随着后续优化的过程中算法对路径不断进行“拆毁”和“重组”,这些客户可能最终也没有机会落在最好的位置上,这样的客户可以称之为边界点。同时,处于路径分布边界的也叫做边界点。如图 4-11 所示,颜色标灰的客户点 4 和客户点 9 就是两个边界点,把他们放到相邻的车辆上更合适(见图 4-11 中的 (b) 图)。于是,我们有意在生成初始解时保存每辆车中的边界点,最后再重新安排它们,尝试将她们从原有路径中抽出来,安置到其他路径中(相邻路径)。除此之外,后续优化过程还会对整个解执行一次扰动,扰动过程由最短路径移除算子和路径内重定位算子组成,以期进一步减少车辆数目和行驶距离。

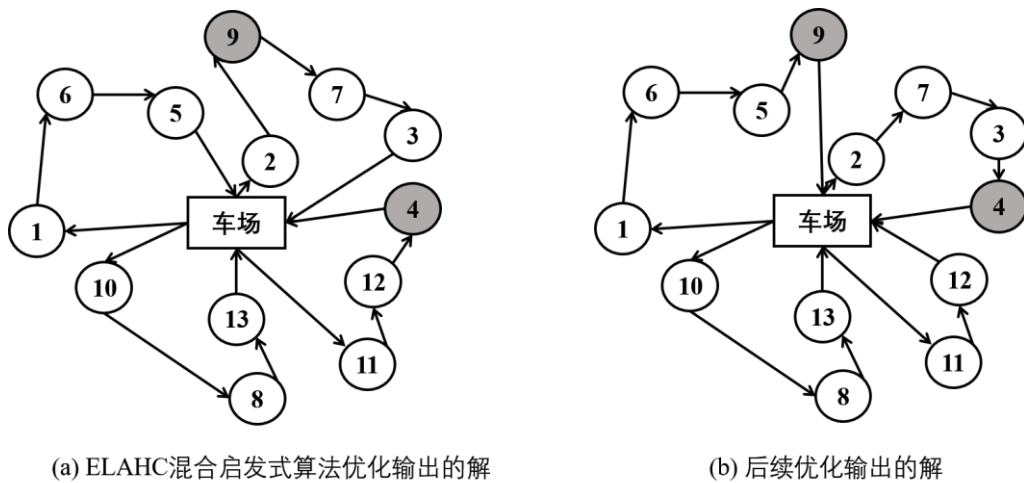


图 4-11 后续优化说明图

4.6 本章小结

本章针对 VRPSPDTW 问题的求解方案进行了详细的描述,首先介绍了算法的大体框架并概述了算法的组成部分,然后详细介绍了算法的每一个步骤。在生成初始解的方法中,本章针对基础扫描算法的不合理性,提出了一种改进型扫描算法来生成更高质量的初始解,接着对主导的混合启发式算法进行了重点阐述并给出了整体算法的伪代码,本章指出基础 LAHC 算法的不足,利用其天然维持的短期存储结构的特征,设计了一种弥补机制来加强算法在迭代后期的搜索能力。算法中的候选解由七种邻域算子产生,分别包括五种基本算子和两种目标导向型算子。除此之外,本章探讨了搜索深度和搜索广度的平衡问题,接着引入强化学习邻域的 MAB 智能选择算法来解决,在合理的时间内最大化搜索的收益。最后,针对性地为扫描生成初始解的过程设计了一种后续优化过程,修复初始解中的边界点,配合两个扰动算子,进一步尝试减少车辆数目和行驶距离。

第五章 仿真实验与应用实例

5.1 引言

第三章给出了本文所研究的 VRPSPDTW 问题的数学模型,第四章详细介绍了本文针对该模型所提出的解决方案,为了验证该方案(h-LAHC 算法)的可行性和高效性,本章将基于公开的基准实例进行仿真实验并基于真实案例进行实践。接下来,我们首先介绍实验的环境以及实验中使用的公共基准实例,然后将 h-LAHC 算法的运行结果与当前权威文献的结果(分别为 p-SA^[50]和 ALNS-PR^[52])进行比较,以证实 h-LAHC 算法的高效性。此外,对 h-LAHC 的重要组成部分进行了分析。最后,本章还将 h-LAHC 方案应用在 A 公司的真实物流订单上,进一步验证了该方案的现实意义。

5.2 仿真实验

5.2.1 实验环境和基准数据集

本文用 Java 这种跨平台的语言实现了上述求解算法,并且所有测试用例都运行在 Intel(R) Core(TM) i5-9700 处理器和 16G RAM 的主机上。

VRPSPDTW 的国际标准数据集^[49]是由 Wang 和 Chen 通过改造 Solomon 基准实例而共同提出的^[5],一共包含 9 个小规模实例和 56 个中等规模实例。若以时间窗的宽窄为分类依据,可以分为 1 类和 2 类,1 类表示客户要求的时间窗比较窄,而二类表示时间窗比较宽;若以客户点的分布为分类依据则可以分为 R 类, C 类和 RC 类, R 类表示客户点的地理分布比较随机和分散, C 类表示客户点的地理分布是聚集的,而 RC 类则是上述两种类型的混合。关于数据实例的分类和实例命名的规则如表 5-1 所示。

表 5-1 基准数据集分类及命名规则

按客户点分布分类	R 类	客户点分布随机
	C 类	客户点分布聚集
	RC 类	客户点分布混合
按时间窗分类	1 类	时间窗窄,车辆载重小
	2 类	时间窗宽,车辆载重大

举例说明实例命名规则

实例命名 = 客户点分布类名+dp+时间窗类名+实例索引

rcdp102: 客户点分布是 rc 类,时间窗是 1 类的第 2 个实例

除了 Wang 和 Chen 的 65 个中小规模的实例外, Wang^[50]在其之后扩展了 30 个包含 200、400、600、800 和 1000 个客户点的大规模实例,但其中的 RC2_8_1 和 RC2_10_1

这两个实例不可用，因此，我们的实验涵盖了由这些实例组成的小型、中型和大型规模数据实例共 93 个。

5.2.2 求解参数设置

在进行实验之前，我们首先进行了试验性实验来对所有参数进行调整，以确保算法能够在合理的时间范围内收敛，并且能够得到高质量的解。表 5-2 展示了这些参数。

表 5-2 参数设置

参数	描述	值
I_{max}	最大的迭代次数	100000
r	收敛率	0.6
l	重启位置的索引比例	4/5
L_h	L_list 的长度	600
α	公式 4-1 的第一个权重	0.9
β	公式 4-1 的第二个权重	0.1
C	平衡因子	0.01
f	最短路径移除算子执行的频率	每 50 轮迭代执行一次

5.2.3 实验结果对比与分析

本文使用 h-LAHC 算法解决 VRPSPDTW 问题，并将运行 10 次的最优结果与目前主流的 p-SA^[50]和 ALNS-PR 算法^[52]进行对比，以验证 h-LAHC 算法的有效性。下面的结果表中，我们主要对比两个优化目标，一个是车辆数目（NV，单位：辆），另一个是总行驶距离（TD，单位：无）。由于所使用的机器设备和编程语言不同，本文不将运行时间（CT，单位：秒）作为主要的比较项，展示出来只为表明，该算法能够在合理的时间内对各种规模的实例进行求解。表结构中的第一列表示数据集中每个实例的名称，命名规则已在 5.2.1 节给出。第二、三、四列分别表示 p-SA、ALNS-PR 和 h-LAHC 算法求解 VRPSPDTW 问题的最优结果。最后两列（以“偏差 1”和“偏差 2”表示）分别展示了 h-LAHC 算法的计算结果和 p-SA 算法以及 ALNS-PR 算法之间的偏差，车辆数目上的偏差以整数表示，而行驶距离上的偏差以百分比表示。

最后一行统计了 h-LAHC 算法比其他算法所得到的较好（+）、近似相等（≈）和较差（-）结果的数量。“近似相等”是指在车辆数量完全相同的情况下，两个解在距离上的百分比偏差小于 5%。

(1) 小规模问题上的实验结果

小规模数据集包含 10、25 和 50 个客户点。表 5-3 列出了每种算法在小规模数据集上的最优实验结果，这三种算法中，ALNS-PR 没有报告结果，因此不在表中显示。从表

5-3 中可以看出, h-LAHC 在所有 (100%) 小规模实例上都能达到与 p-SA 相同或者更好的结果, 在 rcdp2501 这个实例上, h-LAHC 算法得到的解在距离上优于 p-SA。

表 5-3 小规模数据实例上的结果汇总表

数据实例	p-SA 的最优结果			h-LAHC 的最优结果			偏差	
	NV	TD	CT	NV	TD	CT	NV	TD(%)
rcdp1001	3	348.98	1	3	348.98	1	0	0.00
rcdp1004	2	216.69	1	2	216.69	1	0	0.00
rcdp1007	2	310.81	1	2	310.81	1	0	0.00
rcdp2501	5	552.21	2	5	551.05	3	0	-0.21
rcdp2504	4	473.46	1	4	473.46	1	0	0.00
rcdp2507	5	540.87	2	5	540.87	2	0	0.00
rcdp5001	9	994.7	14	9	994.7	24	0	0.00
rcdp5004	6	725.59	16	6	725.59	32	0	0.00
rcdp5007	7	810.04	15	7	810.04	39	0	0.00
统计(+/-)								1/8/0

(2) 中规模问题上的实验结果

中规模数据实例最先由 CoGA^[49]解决, 但是后来的 p-SA 和 ALNS-PR 算法已经刷新了绝大多数结果, 表 5-4 和表 5-5 分别展示了每种算法在 1 类数据集和 2 类数据集上的实验结果。先来看表 5-4, h-LAHC 算法在 1 类数据集中取得了很好的成绩, 与 p-SA 相比, h-LAHC 在 26 个 (89%) 实例上得到了较好的结果, 在 3 个 (11%) 实例上得到了与 p-SA 近似相等的解, 并且没有较差的解。更准确地说, 这 29 个实例中有 12 个实例在车辆数目方面得到了优化 (第一优化目标), 而在其余 17 个实例上的车辆数与 p-SA 算法是相同的。对于具有相同车辆数的 17 个实例, h-LAHC 算法改善了 14 个实例的行驶距离, 剩余 3 个实例上获得了与之相同或近似相等的行驶距离。还可以从表中的数据中发现, 在某些实例中, 当车辆数目得到优化时, 行驶距离会有一定程度的上升, 这是正常现象, 车辆数目作为第一优化目标, 它的减少可能会牺牲在行驶距离方面的优化。

与 ALNS-PR 算法得出的结果相比, h-LAHC 的表现也很好, 优化了 29 个实例中的 14 个 (49%) 实例, 并且总共减少了 2 辆车。另外, 在 10 个 (34%) 实例上 h-LAHC 算法取得了与 ALNS-PR 算法相等或近似相等的解, 在 4 个实例上 (17%) 的表现比 ALNS-PR 更差。整体来看, h-LAHC 在 1 类中规模问题上的表现是比 p-SA 和 ALNS-PR 算法都要好的。

表 5-5 展示的是每种算法在 2 类中规模数据集上的表现。与 p-SA 算法相比, 在 27 个实例中, h-LAHC 算法在 18 个 (67%) 实例中取得了更好的解, 在 9 个 (33%) 实例中得到了相同或近似结果, 没有更差的解。具体来说, h-LAHC 算法在 27 个实例中总共减少了 1 辆车, 剩下 26 个实例的车辆数与 p-SA 算法是一样的。h-LAHC 算法在优化行

驶距离方面也得到 17 个较好的结果。在 9 个相同或近似的解中,有 7 个解与 p-SA 算法的结果相同,其余 2 个与 p-SA 算法得到的结果的偏差约为 2%。与 ALNS-PR 算法相比,在大多数实例下 (89%),h-LAHC 算法得到的结果与 ALNS-PR 算法的结果相等或者近似相等,只有三个更差的解。

总体而言,h-LAHC 算法能够有效地解决所有中等规模的问题。与 p-SA 算法相比,h-LAHC 算法在 100%的实例上都能得到较好或者近似相等的结果,而与 ALNS-PR 算法相比,h-LAHC 算法在 86%的实例上能够得到更好或者相近的结果。特别地,h-LAHC 算法在解决时间窗相对较窄的 1 类问题方面表现突出,而 1 类问题正好对应于目前城市配送中,客户对交付时间的要求比较严格的这一场景。

表 5-4 1 类中规模数据实例上的结果汇总表

数据实例	p-SA 的最优结果			ALNS-PR 的最优结果			h-LAHC 的最优结果			偏差 1		偏差 2	
	NV	TD	CT	NV	TD	CT	NV	TD	CT	NV	TD%	NV	TD%
rdp101	19	1660.98	43	19	1650.8	22.86	19	1655.04	54	0	-0.36	0	0.26
rdp102	17	1491.75	29	17	1486.12	20.89	17	1489.92	42	0	-0.12	0	0.26
rdp103	14	1226.77	41	13	1297.01	17.83	14	1235.87	38	0	0.74	1	-4.71
rdp104	10	1000.65	45	10	984.81	28.4	10	1004.84	36	0	0.42	0	2.03
rdp105	14	1399.81	45	14	1377.11	17.58	14	1396.88	32	0	-0.21	0	1.44
rdp106	12	1275.69	37	12	1252.03	27.54	12	1281.08	45	0	0.42	0	2.32
rdp107	11	1082.92	35	10	1121.86	18.79	10	1102.38	41	-1	1.80	0	-1.74
rdp108	10	962.48	41	9	965.54	20.6	10	955.53	39	0	-0.72	1	-1.04
rdp109	12	1181.92	46	11	1194.73	16.08	11	1178.65	46	-1	-0.28	0	-1.35
rdp110	11	1106.52	45	10	1148.2	19.13	10	1201.47	50	-1	8.58	0	4.64
rdp111	11	1073.62	41	10	1098.84	22.15	10	1096.74	49	-1	2.15	0	-0.19
rdp112	10	966.06	43	9	1010.42	28.63	9	1104.34	49	-1	14.31	0	9.30
cdp101	11	992.88	36	11	976.04	19.3	10	833.24	42	-1	-16.08	-1	-14.63
cdp102	10	955.31	38	10	941.49	28.11	10	832.89	46	0	-12.81	0	-11.53
cdp103	10	958.66	34	10	892.98	48.03	10	880.56	54	0	-8.15	0	-1.39
cdp104	10	944.73	35	10	871.4	46.51	9	961.14	52	-1	1.74	-1	10.30
cdp105	11	989.86	37	10	1053.12	15.97	10	828.94	46	-1	-16.26	0	-21.29
cdp106	11	878.29	37	10	967.71	17.36	10	895.68	59	-1	1.98	0	-7.44
cdp107	11	911.90	41	10	987.64	18.06	10	829.7	49	-1	-9.01	0	-15.99
cdp108	10	1063.73	39	10	932.88	18.22	10	862.4	41	0	-18.93	0	-7.56
cdp109	10	947.90	21	10	910.95	38.45	10	860.42	38	0	-9.23	0	-5.55
rcdp101	15	1659.59	47	14	1776.58	10.59	15	1644.8	48	0	-0.89	1	-7.42
rcdp102	13	1522.76	41	12	1583.62	19.19	13	1519.78	44	0	-0.20	1	-4.03
rcdp103	11	1344.62	45	11	1283.52	29.08	11	1256.74	52	0	-6.54	0	-2.09
rcdp104	10	1268.43	47	10	1171.65	22.57	10	1173.66	52	0	-7.47	0	0.17
rcdp105	14	1581.54	46	14	1548.96	16.95	14	1571.49	55	0	-0.64	0	1.45
rcdp106	13	1418.16	41	12	1392.47	20.47	12	1401.52	47	-1	-1.17	0	0.65
rcdp107	11	1360.17	35	11	1255.06	21.22	11	1284.65	46	0	-5.55	0	2.36
rcdp108	11	1169.57	38	10	1198.36	19.52	10	1157.76	41	-1	-1.01	0	-3.39
统计(+/-)										26/3/0		14/10/5	

表 5-5 2 类中规模数据实例上的结果汇总表

数据实例	p-SA 的最优结果			ALNS-PR 的最优结果			h-LAHC 的最优结果			偏差 1		偏差 2	
	NV	TD	CT	NV	TD	CT	NV	TD	CT	NV	TD%	NV	TD%
rdp201	19	1660.98	43	19	1650.8	22.86	18	1618.42	24	-1	-2.56	-1	-1.96
rdp202	17	1491.75	29	17	1486.12	20.89	17	1466.01	25	0	-1.73	0	-1.35
rdp203	14	1226.77	41	13	1297.01	17.83	14	1246.46	36	0	1.61	1	-3.90
rdp204	10	1000.65	45	10	984.81	28.4	10	981.77	32	0	-1.89	0	-0.31
rdp205	14	1399.81	45	14	1377.11	17.58	14	1399.81	25	0	0.00	0	1.65
rdp206	12	1275.69	37	12	1252.03	27.54	12	1308.28	30	0	2.55	0	4.49
rdp207	11	1082.92	35	10	1121.86	18.79	10	1119	22	-1	3.33	0	-0.25
rdp208	10	962.48	41	9	965.54	20.6	10	940	27	0	-2.34	1	-2.65
rdp209	12	1181.92	46	11	1194.73	16.08	11	1178.65	30	-1	-0.28	0	-1.35
rdp210	11	1106.52	45	10	1148.2	19.13	10	1203.72	28	-1	8.69	0	4.75
rdp211	11	1073.62	41	10	1098.84	22.15	10	1096.74	25	-1	2.15	0	-0.19

表 5-5 2 类中规模数据实例上的结果汇总表 (续)

cdp201	3	591.56	86	3	591.56	24.37	3	591.56	101	0	0.00	0	0.00	
cdp202	3	591.56	91	3	591.56	46.09	3	591.56	138	0	0.00	0	0.00	
cdp203	3	591.17	88	3	591.17	44.19	3	600.21	148	0	1.53	0	1.53	
cdp204	3	594.07	90	3	590.6	51.76	3	593.11	156	0	-0.16	0	0.42	
cdp205	3	588.88	90	3	588.88	36.35	3	588.88	128	0	0.00	0	0.00	
cdp206	3	588.49	88	3	588.49	36.39	3	588.49	116	0	0.00	0	0.00	
cdp207	3	588.29	85	3	588.29	39.83	3	588.29	146	0	0.00	0	0.00	
cdp208	3	599.32	83	3	588.32	34.47	3	589.14	145	0	-1.70	0	0.14	
rcdp201	10	1063.73	39	10	932.88	18.22	4	1432.03	98	0	-5.40	0	1.78	
rcdp202	10	947.90	21	10	910.95	38.45	4	1222.8	123	0	-3.96	1	-13.56	
rcdp203	15	1659.59	47	14	1776.58	10.59	3	1057.36	105	0	-5.89	0	0.64	
rcdp204	13	1522.76	41	12	1583.62	19.19	3	855.96	132	0	-4.59	0	7.20	
rcdp205	11	1344.62	45	11	1283.52	29.08	4	1362.51	117	0	-0.63	0	5.00	
rcdp206	10	1268.43	47	10	1171.65	22.57	3	1155.38	138	0	-0.99	0	0.79	
rcdp207	14	1581.54	46	14	1548.96	16.95	3	1110.16	156	0	1.86	0	4.55	
rcdp208	13	1418.16	41	12	1392.47	20.47	3	862.89	127	0	0.00	0	4.20	
统计(+/-)											18/9/0		0/24/3	

(3) 大规模问题上的实验结果

p-SA 算法首先解决了包含 200、400、600、800 和 1000 个客户点的大型实例。随后, ALNS-PR 尽管在算法设计上比较复杂, 计算量也较大, 但它能够得到更好的结果。由于 Wang^[50]使用随机方法生成了 VRPSPDTW 问题中客户点的提货需求, 为了与他们的结果进行比较, 我们联系了作者, 得到了他们为 VRPSPDTW 问题生成的 30 个包含 200-1000 个客户点的大规模实例。然而, 实例 RC2_8_1 和 RC2_10_1 已经被证实是不可行的两个实例^[52], 因此, 本文在 28 个可用实例上进行了实验, 表 5-6 报告了所有实验结果。

表 5-6 大规模数据实例上的结果汇总表

数据实例	p-SA 的最优结果			ALNS-PR 的最优结果			H-LAHC 的最优结果			偏差 1		偏差 2		
	NV	TD	CT	NV	TD	CT	NV	TD	CT	NV	TD%	NV	TD%	
C1_2_1	21	3169.52	62	20	2846.2	55.54	20	2728.62	123	-1	-13.91	0	-4.13	
R1_2_1	22	5083.39	89	20	4849.8	113.46	21	5191.14	160	-1	2.12	1	7.04	
RC1_2_1	20	3865.18	81	19	3652.18	77.05	19	3766.19	207	-1	-2.56	0	3.12	
C1_4_1	42	8135.35	147	40	7533.03	220.13	40	7523.68	465	-2	-7.52	0	-0.12	
R1_4_1	42	12202.62	237	40	10671.7	537.22	39	11820.61	874	-3	-3.13	-1	10.77	
RC1_4_1	40	10036.82	193	38	9772.56	401.1	38	8979.8	896	-2	-10.53	0	-8.11	
C1_6_1	69	19720.65	257	63	15594.21	765.79	66	16600.52	1426	-3	-15.82	3	6.45	
R1_6_1	62	25729.28	581	59	22306.17	1211.12	59	22027.59	1625	-3	-14.39	0	-1.25	
RC1_6_1	60	20535.26	733	57	19679.75	946.06	56	20400.75	1560	-4	-0.66	-1	3.66	
C1_8_1	88	32801.92	1054	82	27035.71	1495.17	87	29779.13	1796	-1	-9.22	5	10.15	
R1_8_1	93	51949.49	1869	80	39348.17	3082.84	80	44201.21	3358	-13	-14.92	0	12.33	
RC1_8_1	88	32801.92	1620	75	38431.09	2234.79	75	35230.88	3042	-13	7.40	0	-8.33	
C1_10_1	110	52328.78	2418	102	44764.64	2253.4	108	52116.02	2984	-2	-0.41	6	16.42	
R1_10_1	115	77993.35	4539	100	58912.62	4651.74	100	61575.36	5421	-15	-21.05	0	4.52	
RC1_10_1	102	66883.49	3483	93	63953.66	4325.54	92	53024.33	4698	-10	-20.72	-1	-17.09	
C2_2_1	6	1972.97	112	6	1931.44	194.73	6	1942.86	244	0	-1.53	0	0.59	
R2_2_1	5	4372.17	295	5	4042.67	186.02	5	4298.17	256	0	-1.69	0	6.32	
RC2_2_1	4	2662.75	216	4	2021.49	338.41	4	2389.31	563	0	-10.27	0	18.20	
C2_4_1	14	5085.08	279	12	4144.84	776.51	13	6237.85	965	-1	22.67	1	50.50	
R2_4_1	9	14119.64	735	9	8952.24	506.62	9	9512.47	921	0	-32.63	0	6.26	
RC2_4_1	13	7229.22	791	12	6621.94	681.44	13	7253.77	993	0	0.34	1	9.54	
C2_6_1	20	9509.15	926	18	7830.16	1652.55	18	7842.45	1875	-2	-17.53	0	0.16	
R2_6_1	13	27294.11	2439	13	17459.41	1558.88	13	20412.08	1993	0	-25.21	0	16.91	
RC2_6_1	20	22837.36	2860	16	12693.19	2607.91	18	14551.45	3816	-2	-36.28	2	14.64	
C2_8_1	27	14573.93	3636	24	11759.05	3022.47	24	12966.54	4012	-3	-11.03	0	10.27	
R2_8_1	19	48611.63	7663	18	27270.04	3429.95	16	35650.49	6941	-3	-26.66	-2	30.73	
RC2_8_1	-	-	-	-	-	-	-	-	-	-	-	-	-	
C2_10_1	33	23981.11	6529	30	17088.5	5009.45	31	16893.24	7013	-2	-29.56	1	-1.14	
R2_10_1	22	67441.51	21379	22	42117.48	5604.7	21	53279.82	9124	-1	-21.00	-1	26.50	
RC2_10_1	-	-	-	-	-	-	-	-	-	-	-	-	-	
统计(+/-)											27/1/0		10/3/15	

与 p-SA 算法相比, h-LAHC 算法在 28 个实例中改进了 27 个(97%)结果, 并且在 22 个实例中总共减少了 86 辆车, 这可以说是一个非常大的进步, 表明随着客户点的增多和问题规模的增大, h-LAHC 算法在优化车辆数目方面的效果会更加显著。在 5 个车辆数目相等的实例中, h-LAHC 算法得到的结果相比 p-SA 算法的结果在很大程度上减少了整体运输距离, 减少的比例最高达到 36%。而与 ALNS-PR 算法相比, h-LAHC 算法也具有很强的竞争性, 在 14 个(50%)实例中, h-LAHC 算法得到了较好的或近似的结果, 总共减少了 6 辆车。与此同时还可以发现, 在大规模问题中, h-LAHC 算法也在时间窗相对狭窄的 1 类问题中表现得更加优秀, 减少车辆数目的效果更加明显, 而 ALNS-PR 算法更适合于时间窗相对宽裕, 车辆载重更大的 2 类问题。

5.2.4 对重要组件的实验分析

为了进一步探究本文提出的 h-LAHC 算法的高效性和其中的原因, 本节针对 h-LAHC 算法的重要组件进行验证分析。这些组件分别是改进型扫描生成算法、加强型延迟爬山算法以及多臂老虎机智能选择算法。

(1) 改进型扫描生成算法的效果

由于本文提出的 VRPSPDTW 问题的求解方案是基于单解的, 为了生成一个高质量的初始解, 让初始解中车辆上的客户点相对更集中, 本文将客户点到车场的距离作为另一个因素加入到基本扫描算法中, 和极角因素一起配合, 以决定客户点的插入次序。为了验证改进型扫描生成算法生成初始解的效果, 本文在中等规模数据实例的每种类型中选取前三个实例进行验证实验, 分别使用基础扫描生成算法和改进型扫描生成算法获得初始解。由于基准扫描线的角度是随机选取的, 本文在同等条件分别运行两种算法 10 次再取平均结果, 对比结果如表 5-7 所示, 可以看出, 在 18 个测试实例中, 改进型扫描生成算法的在 14 个(78%)实例上的到初始解的质量都要优于基础扫描算法, 证明该方法在生成 VRPSPDTW 问题的初始解方面具有积极的作用, 也为后续的混合优化算法提供了一个良好的开端。

表 5-7 基础扫描算法和改进型扫描算法生成初始解的对比结果

数据实例	基础扫描算法		改进型扫描算法		偏差	
	NV	TD	NV	TD	NV	TD(%)
cdp101	10	828.94	10	828.94	0	0.00
cdp102	11	925.61	10	857.07	-1	-7.40
cdp103	11	1154.9	11	1091.2	0	-5.51
rdp101	34	2423.3	33	2383.2	-1	-1.65
rdp102	27	1989.5	27	1983.1	0	-0.32

表 5-7 基础扫描算法和改进型扫描算法生成初始解的对比结果（续）

rdp103	18	1612.5	18	1567.5	0	-2.79
rcdp101	22	2279.7	21	2220.6	-1	-2.59
rcdp102	20	2207.6	20	2030.2	0	-8.04
rcdp103	16	1912.2	15	1787.5	-1	-6.52
cdp201	4	838.29	4	875.14	0	4.40
cdp202	4	927.07	4	911.29	0	-1.70
cdp203	5	915.04	4	1015.3	-1	10.96
rdp201	5	1445.6	5	1656.2	0	14.56
rdp202	5	1485.2	5	1391.9	0	-6.28
rdp203	5	1316	5	1192.8	0	-9.36
rcdp201	5	2074.9	5	1976.2	0	-4.76
rcdp202	5	1553	5	1531.6	0	-1.38
rcdp203	4	1267.8	4	1417.6	0	11.81
统计(+/-)		14/1/3				

(2) 加强型延迟爬山算法的效果

本文在基础 LAHC 算法中添加了一个扰动机制帮助算法在迭代后期跳出局部最优，防止算法过早停滞。为了验证这一机制的有效性，本文选取 RC 类型的大规模数据实例 RC1_2_1 进行两组实验，选取 RC 类型是因为混合类型在所有实例中更具有代表性，大规模数据实例是因为该情况下的迭代次数相对较多，能够更好的展示算法的收敛情况。

1) 基础 LAHC 算法（BLAHC）与加强型 LAHC 算法（ELAHC）对比

分别将 BLAHC 算法和 ELAHC 算法应用于上述的测试用例，实验三次取平均结果，观察算法是如何随着迭代次数的增加而收敛的。对比结果如图 5-1 所示。其中，横坐标表示迭代次数，纵坐标表示车辆数目或者行驶距离，图中橙色的线代表 BLAHC 算法，蓝色的线代表 ELAHC 算法，用红色圆圈标记的部分是指算法跳出局部最优解从而获得重新启动的机会。从图中可以看出，随着迭代次数的增加，两个算法在车辆数目和整体行驶距离方面都会逐渐收敛，直到算法停止。两个算法的前期运行曲线的走势是基本一致的，BLAHC 算法在迭代后期基本上是一条直线，而 ELAHC 算法在迭代后期有一定的起伏，这个起伏是由本文为 BLAHC 算法设计的扰动机制产生的。可见，随着算法的继续运行，ELAHC 算法拥有跳出局部最优解的能力，虽然这将使当前的解决方案暂时变得更糟，但它可能使算法朝着更好的方向移动，进而收敛到下一个更好的局部最优解。值得一提的是，ELAHC 的本质还是基于随机原理的智能启发式算法，因此，并不是每一次执行中的扰动过程都能让算法跳出局部最优解并且成功找到一个更好的搜索方向。从图中还可以发现，扰动机制虽然能够在一定程度上帮助算法跳出局部最优解，但这种重新启动的代价就是迭代次数会相应增加，运行时间也会更长，尽管如此，ELAHC 算法还是能在合理的时间范围内换取了更高的精度。

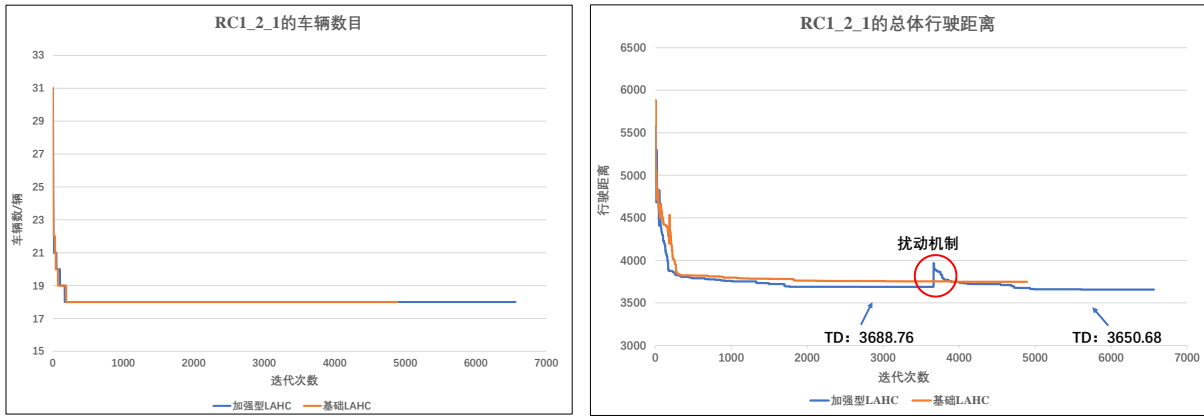


图 5-1 BLAHC 和 ELAHC 在 RC1_2_1 上的收敛状况对比

2) 增加迭代次数和扰动机制对比

为了证实 ELAHC 算法所得的结果不是单纯增加迭代次数得到的, 本文首先在 RC1_2_1 实例上执行了三次 ELAHC 算法, 获得了平均迭代次数。接着, 让 BLAHC 算法也运行到与之相同的迭代次数, 观察各自的收敛情况, 绘制成图 5-2 所示的曲线。可以发现, 人为地增加算法的迭代次数也有机会让算法跳出局部最优解, 但这个比变化幅度仍然比不上扰动机制, 而在大部分情况下, 单纯增加迭代次数无法让算法跳出局部最优解, 因此算法后期的走势曲线仍然会呈一条直线。

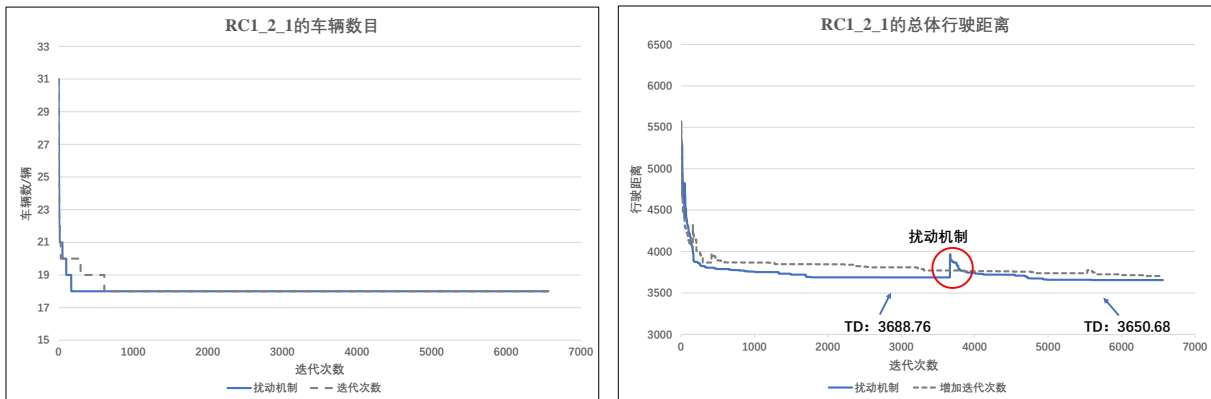


图 5-2 ELAHC 的加强机制和增加迭代次数在 RC1_2_1 上的收敛状况对比

通过以上两个实验, 可以证明 ELAHC 算法能够更高精度地解决 VRPSPDTW 问题, 但也会在合理的范围内增加一定的运行时间。

(3) 多臂老虎机智能选择算法的效果

本文为 ELAHC 算法设计了多种邻域算子, 并且引入 MAB 智能选择算法来最大化邻域算子的作用, 达到探索深度和探索广度之间的平衡。为了验证 MAB 算法的有效性, 本文在中等规模的数据实例中选取每个类型的前三个实例进行实验, 在其他条件相同的情况下, 分别执行 MAB 智能选择邻域算子的方案、随机选择邻域算子的方案以及邻域

算子全部使用的方案, 每种方案运行三次并取平均值, 将结果记录在表 5-8 中。从表中可以看出, 随机选择邻域算子的方案得出的结果在精度方面最差, 在时间方面和使用 MAB 智能选择方案差不多, 而邻域算子全部使用的方案在精度方面和 MAB 智能选择邻域算子的方案相当, 但运行时间明显增加, 由于 ELAHC 算法本身是牺牲了一定时间换取了精度, 因此, 邻域算子在每轮迭代中全部将使得整体算法更消耗时间, 从这一层面上来看, MAB 智能选择算法可以最大化搜索过程中的收益, 使得算法能够在合理的时间范围内获取较高精度的解。

表 5-8 邻域选择的三种方案得到的最终结果

数据实例	MAB			随机选择			全部邻域		
	NV	TD	CT	NV	TD	CT	NV	TD	CT
rdp101	19	1665.36	56	19	1989.23	54	19	1667.96	64
rdp102	17	1491.58	47	18	1978.79	40	17	1490.63	77
rdp103	14	1251.42	35	14	1633.37	38	14	1245.09	67
cdp101	10	833.24	46	11	900.07	47	10	833.24	61
cdp102	10	841.93	52	10	877.49	50	10	843.38	68
cdp103	10	890.84	58	10	970.74	51	10	907.23	63
rcdp101	15	1649.67	40	15	1718.94	42	15	1649.59	70
rcdp102	13	1522.76	58	14	1532.95	56	13	1524.76	67
rcdp103	11	1273.25	47	12	1384.84	48	11	1269.14	64
rdp201	4	1289.12	123	4	1711.96	121	4	1284.12	227
rdp202	3	1202.27	134	4	1151.11	128	4	1150.31	263
rdp203	3	979.96	128	4	1143.44	119	3	982.06	245
cdp201	3	591.56	104	4	761.82	112	3	591.56	284
cdp202	3	591.56	132	3	870.09	130	3	591.56	254
cdp203	3	624.41	124	4	931.58	113	3	645.11	267
rcdp201	4	1513.72	117	4	1667.18	116	4	1500.54	234
rcdp202	4	1298.87	136	4	1775.29	135	4	1271.56	251
rcdp203	3	1119.54	142	4	1596.85	139	3	1160.16	227

5.3 应用实例

为了验证本文针对 VRPSPDTW 问题提出的求解方案在实际问题中的可行性, 本课题组从 A 物流公司的同城配送需求出发, 为现实物流订单提供智能车货调度的方案。

5.3.1 A 公司简介

A 公司 2015 年成立于广州, 是某大型物流集团的核心子公司之一, 是将互联网融入传统物流的行业先进企业。提供的产品和服务以互联网和信息技术为支撑, 依靠自身和母公司丰富的物流服务经验, 结合社会优势资源, 为客户提供透明化、信息化、智能化的一站式公路运输服务。A 公司在全国 130 多个城市建立了运营平台, 形成了一个可以覆盖全国物流业务的运作网络, 并且未来还将继续迎合市场的变化和客户需求日益增长的需求, 继续扩大服务平台的建设。

A 公司在物流运输方面的主营业务包含订单同步、智能调度、预约装货、自动发车、

在途跟踪、自动到货、回单识别、线上结算、售后等等。为此，该公司自主研发了互动化、可视化、透明化的一站式智慧物流监控系统。

5.3.2 A 公司的同城配送需求

随着城市发展和互联网供应链新模式的出现，加上目前经济活动逐渐以消费者为中心，末端物流作为直接和目标客户对接的配送方式，也逐渐成为各大企业竞争的利器。一方面，同城配送方便同城客户快速收发货物，另一方面，市内的车队和其他物流资源都能更好地管理。因此，A 公司的服务路线除了目前的省际路线和省内路线外，还想继续发展市内路线来增加企业竞争力。目前 A 公司已经接入了市内配送的订单，但是所有排线操作都是由人工完成的，也就是说，车型的选择、货物的匹配和路径的规划都是由人工主观判断的，由于当前 A 公司接入的订单量不算多，人工排线的方式是可行的，但如果未来订单量增多，人工排线将会耗费大量人力和时间，并且人工排线的方案在物流成本上可能不是最优的方案。因此，A 公司亟需一个智能排线工具，既能帮助它们智能化地完成上述关键步骤，又能弥补因人工判断的偏差而带来的成本损失。

A 公司目前在同城配送方面的运营特点整理如下

1) 去中心化的车场

在 A 公司的物流配送模式也有车场这种物流配送中心的概念，他们将大型的取货点称之为据点，比如目前的广州市白云机场物流中心，车辆每天会从该据点出发，前往城市内的相应位置进行货物配送。

2) 取送货一体模式

A 公司的配送模式并不是单纯的取货或者单纯送货，由于除了大型据点以外还存在其他小规模甚至个体经营的取货点，因此，在服务过程中，车辆可以直接送货，也可以先去取货点取货（相当于商户），再去相应的送货点送货（相当于客户），因此，该模式下的取送货属于混合取送货，不严格属于的同时取送货。

3) 确定的运输需求

A 公司的配送订单都是静态的，也就是说，同一批订单只会被处理一次，并且在运输之前就已经确定了订单的所有信息，不存在车辆配送过程中临时插入紧急订单的动态情况。一般都是在生成运输计划的前一天把第二天要运输的订单整理出来，并且通过人工或者智能调度的方式为己确定的订单排线，确定使用的车辆以及运输的路线。

4) 严格约束的时间窗

在实际物流运营的过程中，会尽一切努力来达到客户要求的时间窗，如果车辆比双

方约定的最晚时间还要晚到达服务点,可能会使得客户的计划被打乱,从而产生一系列不必要的争执,也会使得物流公司信誉受损。因此,A公司提供的物流服务也严格控制在客户要求的时间窗以内,属于硬时间窗。

5.3.3 应用案例求解

本节使用A公司的真实订单数据来验证本文求解VRPSPDTW问题的方案的实际可行性,统一选择型号为“4.2厢车”的运输车辆类型,其最大载重为300千克,并选取20个客户的订单作为实验数据来进行排线操作,为了保护客户信息,本文为所有订单数据脱敏处理并展示在表5-9中。表中取货点和送货点的数据,是根据真实订单中地理位置的经纬度进行一定处理得到的相对坐标,其中订单号为0的数据没有配送需求,代表的是车辆的出发点,也同时是一个大型据点,因此,有一些订单取货点的位置与据点位置一致。值得注意的是,下文给出的求解结果是基于真实订单数据,而不是脱敏后的订单数据,客户点之间的距离不是简单的欧氏距离,而是基于百度地图服务商提供的公共接口计算而来的。

表 5-9 20 个客户的订单数据

订单号	取货点	送货点	货重(kg)	取货点服务时间(min)	送货点服务时间(min)	取货时间窗	送货时间窗
0	[205, 210]	-	-	-	-	-	-
1	[205, 210]	[201, 105]	85	10	10	8:00-10:00	10:30-15:00
2	[205, 210]	[170, 136]	12	3	3	8:00-10:00	12:50-13:40
3	[205, 210]	[182, 210]	36	5	5	8:00-10:00	14:30-15:30
4	[205, 210]	[177, 145]	55	5	5	8:00-10:00	15:00-17:00
5	[205, 210]	[201, 146]	126	17	17	8:00-10:00	9:30-12:00
6	[205, 210]	[110, 235]	41	10	10	8:00-10:00	14:15-16:00
7	[205, 210]	[198, 207]	88	10	10	8:00-10:00	10:30-14:00
8	[205, 210]	[187, 196]	76	9	9	8:00-10:00	15:00-20:30
9	[215, 199]	[207, 168]	45	5	5	8:00-10:00	16:00-20:00
10	[133, 102]	[185, 124]	25	4	4	7:00-10:45	12:40-14:40
11	[133, 102]	[114, 106]	84	10	10	8:00-10:30	10:50-12:50
12	[144, 187]	[175, 187]	78	10	10	12:50-13:50	14:05-16:50
13	[138, 203]	[195, 122]	66	8	8	9:00-11:30	10:00-11:00
14	[174, 152]	[146, 136]	74	8	8	7:30-9:30	11:00-17:30
15	[141, 136]	[211, 174]	26	2	2	9:30-12:30	10:00-13:00
16	[199, 101]	[190, 169]	53	5	5	9:20-12:00	12:20-14:30
17	[142, 130]	[158, 123]	36	3	3	9:00-12:00	13:00-13:50
18	[193, 200]	[210, 196]	89	10	10	7:00-12:00	12:15-15:30
19	[112, 106]	[126, 140]	51	7	7	11:20-12:50	13:40-14:50
20	[171, 185]	[190, 172]	21	3	3	10:00-13:30	14:00-19:50

在使用本文求解VRPSPDTW问题的方案来求解实际问题之前,首先要将该实际订单数据转换成符合VRPSPDTW问题特征的数据,在VRPSPDTW问题中,车辆会在同一个客户点处同时取送货,也就是说,取货点和送货点的地理位置是同一个,而在A公司的真实订单中,取货点和送货点的地理位置是不同的,因此本文将该订单数据转换成表5-10所示的数据,将每笔订单的取货点和送货点拆开,“+”表示取货点,“-”表示送货点,每个客户点都同时包含取货和送货的需求,其中取货点的送货需求置为0,对应

地，送货点的取货需求为 0。在算法执行的过程中，对应的取货点和送货点会一起执行插入或者移除操作，并且保证取货点在路径中的配送顺序在送货点之前，这样，以订单形式呈现的客户数据即被转换为符合 VRPSPDTW 问题的格式。

表 5-10 订单数据转换为 VRPSPDTW 问题的数据

订单号	客户点	取货重量(kg)	送货重量(kg)	服务时间(min)	服务时间窗
0	[205, 210]	-	-	-	-
1+	[205, 210]	85	0	10	8:00-10:00
1-	[201, 105]	0	85	10	10:30-15:00
2+	[205, 210]	12	0	3	8:00-10:00
2-	[170, 136]	0	12	3	12:50-13:40
3+	[205, 210]	36	0	5	8:00-10:00
3-	[182, 210]	0	36	5	14:30-15:30
4+	[205, 210]	55	0	5	8:00-10:00
4-	[177, 145]	0	55	5	15:00-17:00
5+	[205, 210]	126	0	17	8:00-10:00
5-	[205, 210]	0	126	17	9:30-12:00
6+	[205, 210]	41	0	10	8:00-10:00
6-	[110, 235]	0	41	10	14:15-16:00
7+	[205, 210]	88	0	10	8:00-10:00
7-	[198, 207]	0	88	10	10:30-14:00
8+	[205, 210]	76	0	9	8:00-10:00
8-	[187, 196]	0	76	9	15:00-20:30
9+	[215, 199]	45	0	5	8:00-10:00
9-	[207, 168]	0	45	5	12:40-14:40
10+	[133, 102]	25	0	4	7:00-10:45
10-	[185, 124]	0	25	4	12:40-14:40
11+	[133, 102]	84	0	10	8:00-10:30
11-	[114, 106]	0	84	10	10:50-12:50
12+	[144, 187]	78	0	10	12:50-13:50
12-	[175, 187]	0	78	10	14:05-16:50
13+	[138, 203]	66	0	8	9:00-11:30
13-	[195, 122]	0	66	8	10:00-11:00
14+	[174, 152]	74	0	8	7:30-9:30
14-	[146, 136]	0	74	8	11:00-17:30
15+	[141, 136]	26	0	2	9:30-12:30
15-	[211, 174]	0	26	2	10:00-13:00
16+	[199, 101]	53	0	5	9:20-12:00
16-	[190, 169]	0	53	5	12:20-14:30
17+	[142, 130]	36	0	3	9:00-12:00
17-	[158, 123]	0	36	3	13:00-13:50
18+	[193, 200]	89	0	10	7:00-12:00
18-	[210, 196]	0	89	10	12:15-15:30
19+	[112, 106]	51	0	7	11:20-12:50
19-	[126, 140]	0	51	7	13:40-14:50
20+	[171, 185]	21	0	3	10:00-13:30
20-	[190, 172]	0	21	3	14:00-19:50

接下来将本文针对 VRPSPDTW 问题设计的 h-LAHC 算法应用在真实物流数据中，步骤如下：

- 1) 导入车辆信息和订单信息，并且持久化；
- 2) 利用百度地图 API 将订单中每个客户点详细位置转化为可以参与计算的经纬度；
- 3) 将每条订单都拆分成对应的取货点和送货点，转化成 VRPSPDTW 问题的数据格式；
- 4) 将车辆对应的最大载重等关键信息和客户点的经纬度、需求量等关键信息作为算法的输入，执行 h-LAHC 算法进行优化求解；

5) 将输出的配送计划持久化到数据库中。

最终得到的结果如表 5-11 所示，与人工排线的对比如图 5-3 所示，智能排线一共需要使用 3 辆车，总行驶距离为 4577.43 千米，而 A 物流公司人工调度的配送计划中，总共需要 4 辆车，总行驶距离为 5098.32 千米。可见，本文求解 VRPSPDTW 问题的方案能够智能化地解决现实生活中的实际物流配送问题。从表 5-9 中还可以看出，这些客户点的地理分布既有聚集的，又有分散的，并且大部分客户要求的配送时间窗都很窄，所有客户的需求都在同一天内配送完毕，充分表明了本文针对求解 VRPSPDTW 问题提出的 h-LAHC 算法的高效性和实际可行性。

表 5-11 25 个客户配送计划表

运输车辆	客户点	配送路线
1	(1,2,3,4,7)	(0)→(1+)→(2+)→(3+)→(4+)→(7+)→(2-)→(1-)→(7-)→(3-)→(4-)→(0)
2	(5,6,8,9,12,14,17,20)	(0)→(5+)→(6+)→(8+)→(5-)→(9+)→(14+)→(20+)→(17+)→ (14-)→(12+)→(17-)→(20-)→(6-)→(12-)→(8-)→(9-)→(0)
3	(10,11,13,15,16,18,19)	(0)→(11+)→(10+)→(13+)→(13-)→(16+)→(18+)→(11-)→(15+)→ (15-)→(19+)→(10-)→(18-)→(16-)→(19-)→(0)

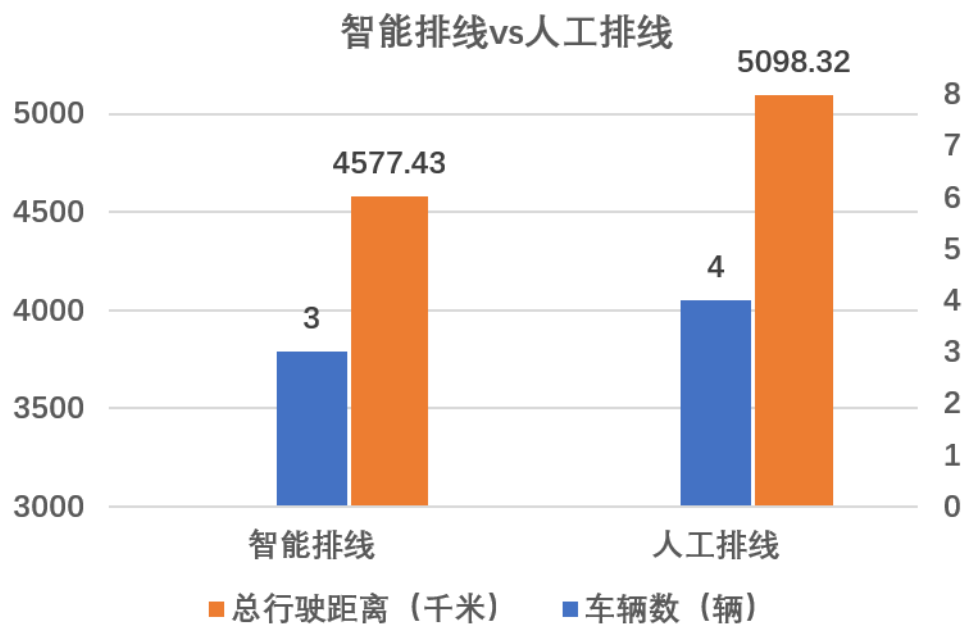


图 5-3 智能排线 vs 人工排线

5.4 本章小结

本章介绍了实验数据和实验环境，在国际公开的 VRPSPDTW 问题的数据集上进行了验证实验，并将本文针对该问题提出的 h-LAHC 算法与最近的两种主流算法 p-SA 和 ALNS-PR 进行了对比，结果显示，本文的求解方案与 p-SA 算法对比，可以在 100%的

实例上得到更好或者与之相近的解，与 ALNS-PR 算法相比，可以在 76% 的实例上得到更好或者与之相近的解，从而验证了本文的求解方案在求解 VRPSPDTW 问题上的高效性。特别地，本文的求解方案在时间窗相对较窄的场景下具有显著的优越性，而这一场景恰好对应于与人们息息相关的城市配送。此外，本章结合了 A 物流公司在同城配送上的实际运营场景，将本文的求解方案成功运用在了 A 公司的实际订单数据上，充分表明了本文求解 VRPSPDTW 问题的方案对现实物流活动具有一定指导意义。

总结与展望

随着电商的蓬勃发展和人们生活质量的提高,加上环境问题受到广泛的关注以及资源再生技术的进步,传统物流行业受到了一次次冲击,物流配送模式也在不断地改革创新,因此,如何在顺应时代发展的前提下更高效地降低物流成本、达到经济效益和环境效益的共同增加、提升顾客们的满意度已经成为各大企业的共同目标。而带时间窗和同时取送货的车辆路径问题作为物流领域的重要抽象,是有望实现经济效益和环境效益双赢的一种复杂的物流配送模式,由于目前学术界对 VRPSPDTW 问题的研究并不多,并且现有的研究各有局限性。因此,本文重点研究了一种高效并且易于实现的方法,在合理的时间范围内求解各个规模下的 VRPSPDTW 问题,主要工作总结如下:

1) 建立 VRPSPDTW 问题的数学模型

本文首先对 VRPSPDTW 问题进行描述,然后用数学符号定义了描述中的参数和变量,结合 VRPSPD 问题和 VRPTW 问题的约束条件,建立了 VRPSPDTW 问题的数学模型。

2) 设计改进型基础扫描算法生成高质量的初始解

本文指出了基础扫描算法的不足,将客户点与车场的距离作为一个重要因素,与极角因素共同配合从而决定客户点的插入次序,从而使得每条路径中的客户点分布相对聚集,得到一个更高质量的初始解,为后续的优化算法提供一个良好的起点。

3) 设计一种扰动机制加强延迟爬山算法的性能

延迟爬山算法是一个近年提出的新颖的智能启发式算法,其规模无关性和参数易调节性已经得到验证,但是该算法在迭代后期容易陷入局部最优,因此,本文借助延迟爬山算法天然维持的短期存储结构的特征,设计了一种扰动机制,沿着原来的优化方向重新激活算法,帮助延迟爬山算法在迭代后期跳出局部最优解,从而有望寻找到下一个更优的解决方案。

4) 设计多臂老虎机智能选择算法解决搜索广度和搜索深度之间的平衡问题

现有成果中仍未解决搜索过程中的平衡困境,因此,本文从强化学习领域中获得启发,设计多臂老虎机算法实现搜索广度和搜索深度之间的平衡,使得整体算法能在合理的迭代次数内获得最大的收益,既能较快收敛又能获得高质量的结果。

5) 将本文的求解方案与最新的主流算法进行对比分析,并将其应用于 A 物流公司的真实订单数据上。

本文将提出的 h-LAHC 算法应用于国际标准数据集中, 并将结果和 p-SA 算法以及 ALNS-PR 算法所得的结果进行了对比, 相比于 p-SA 算法, h-LAHC 算法可以在 100% 的实例上得到更好或者与之相近的解, 而与 ALNS-PR 算法相比, h-LAHC 算法可以在 76% 的实例上得到更好或者与之相近的解, 特别是在时间窗较窄的 1 类问题上, h-LAHC 算法的表现十分优越。此外, 为了验证 h-LAHC 算法的实用性, 本文将 VRPSPDTW 问题与 A 物流公司的同城配送场景结合起来, 采用 h-LAHC 算法成功地求解了 A 公司的真实订单数据, 充分体现了该算法在真实场景下的可用性。

本文研究的 VRPSPDTW 问题虽然可以高效地求解国际标准数据集和 A 公司同城配送场景下的部分订单, 但由于笔者的研究时间有限, 本文的研究工作还有不足之处, 未来的还可以从以下几个方面提高:

1) 考虑多车型

现实生活中的运输车辆种类繁多, 车型和货物之间的对应关系会影响整个物流活动的成本, 本文目前研究的还只是同构车型的问题, 如果把多车型因素考虑进来的话, VRPSPDTW 问题会变得更加复杂, 但也更贴合实际。

2) 考虑动态需求

本文研究的 VRPSPDTW 问题是基于需求确定的场景的, 即订单信息必须在调度之前确定下来, 不允许配送的过程中新插入订单或者撤销原有的订单, 这种场景适合客户源固定、与客户建立了长期合作关系的物流公司, 而在更精细化的配送中, 客户的需求经常动态变化, 随机和不确定因素的增加将会使得问题更加复杂。

3) 考虑算法的移植性

本文提出的求解方案只运用在了 VRPSPDTW 问题中, 其中的改进型扫描生成算法属于传统构造算法, 可能仅仅适用于 VRPSPDTW 问题, 而主导的优化算法属于智能启发式算法, 通常具有通用性, 未来的研究工作中可以尝试将本文的求解方案移植到 VRP 问题的其他变种问题中。

参考文献

- [1] 张云霞. 绿色经济下的物流运输成本控制[J]. 物流工程与管理, 2018, 40(07): 58-60.
- [2] Braekers K, Ramaekers K, Nieuwenhuyse I V. The vehicle routing problem: State of the art classification and review[J]. Computers & Industrial Engineering, 2016, 99(sep.): 300-313.
- [3] Haddadene S R A, Labadie N, Prodhon C. Bicriteria Vehicle Routing Problem with Preferences and Timing Constraints in Home Health Care Services[J]. Algorithms, 2019, 12(8): 152.
- [4] Wu L, He Z, Chen Y, et al. Brainstorming-Based Ant Colony Optimization for Vehicle Routing with Soft Time Windows[J]. IEEE Access, 2019: 1-1.
- [5] Solomon M M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints[J]. Operations Research, 1987, 35(2): 254-265.
- [6] Lübbecke M E, Desrosiers J. Selected Topics in Column Generation[J]. Operations Research, 2005, 53(6): 1007-1023.
- [7] Laporte G. The Vehicle Routing Problem : An overview of exact and approximate algorithms[J]. Eur. J. Oper. Res, 1992, 59(3): 345-358.
- [8] Laporte G, Nobert Y, Desrochers M. Optimal Routing under Capacity and Distance Restrictions[M]. INFORMS, 1985.
- [9] Qureshi A G, Taniguchi E, Yamada T. An exact solution approach for vehicle routing and scheduling problems with soft time windows[J]. Transportation Research Part E Logistics Transportation Review, 2009, 45(6): 960-977.
- [10] Azi N, Gendreau M, Potvin J Y. An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles[J]. European Journal of Operational Research, 2007, 202(3): 756-763.
- [11] B G G-J A, C G D, D G L, et al. A branch-and-price algorithm for the Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows[J]. European Journal of Operational Research, 2010, 206(2): 341-349.
- [12] Baker E K, Schaffer J R. Solution Improvement Heuristics for the Vehicle Routing and Scheduling Problem with Time Window Constraints[J]. American Journal of Mathematical Management Sciences, 2013, 6(3-4): 261-300.
- [13] H. R G, Van, Landeghem. A bi-criteria heuristic for the vehicle routing problem with time windows[J]. European Journal of Operational Research, 1988, 36(2): 217-226.
- [14] Ghoseiri K, Ghannadpour S F. Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm[J]. Applied Soft Computing, 2010, 10(4):

1096-1107.

- [15] Lei H, Laporte G, Guo B. The capacitated vehicle routing problem with stochastic demands and time windows[J]. Computers Operations Research, 2011.
- [16] Vidal T, Crainic T G, Gendreau M, et al. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows[J]. Computers Operations Research, 2013, 40(1): 475-489.
- [17] Yassen E T, Ayob M, Nazri M Z A, et al. Meta-harmony search algorithm for the vehicle routing problem with time windows[J]. Information Sciences, 2015.
- [18] Nalepa J, Blocho M. Adaptive memetic algorithm for minimizing distance in the vehicle routing problem with time windows[J]. Soft Computing, 2015.
- [19] Mehdi N, Abdelmoutalib M, Imad H. A modified ALNS algorithm for vehicle routing problems with time windows[J]. Journal of Physics: Conference Series, 2021, 1743(1): 012029 (10pp).
- [20] Ropke S, Pisinger D. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows[J]. Transportation Science, 2006, 40.
- [21] Konstantakopoulos G D, Gayialis S P, Kechagias E P, et al. A Multiobjective Large Neighborhood Search Metaheuristic for the Vehicle Routing Problem with Time Windows[J]. Algorithms, 2020, 13(10): 243.
- [22] 吴斌, 宋琰, 程晶, et al. 基于密度峰值聚类的 VRPTW 问题研究[J]. 工业工程, 2020, (5).
- [23] 盛虎宜. 生鲜电商物流配送的车辆路径问题研究[D]. 2019.
- [24] 于震. 基于 VRPTW 的 J 公司冷链配送优化方案设计[D]. 东华大学, 2019.
- [25] 徐廷学, 张海军, 付霖宇, et al. 基于量子蚁群算法的 VRPTW 研究[J]. 火力与指挥控制, 2019, 44(08): 36-42.
- [26] 蔚帅, 蒋洪伟. 基于聚类的变邻域模拟退火算法求解 VRPTW[J]. 北京信息科技大学学报, 2020, 35(05): 89-95.
- [27] 马隽, 向晓林, 刘萍. 基于修正遗传算法的含服务费的车辆路径问题[J]. 科学技术与工程, 2017, 17(32): 1-5.
- [28] Min H. The multiple vehicle routing problem with simultaneous delivery and pick-up points[J]. Transportation Research Part A General, 1989, 23(5): 377-386.
- [29] Subramanian A, Uchoa E, Ochi L S. New Lower Bounds for the Vehicle Routing Problem with Simultaneous Pickup and Delivery[C]. International Conference on Experimental Algorithms, 2010.

- [30] Subramanian A, Uchoa E, Pessoa A A, et al. Branch-and-cut with lazy separation for the vehicle routing problem with simultaneous pickup and delivery[J]. Operations Research Letters, 2011, 39(5): 338-341.
- [31] Subramanian A, Uchoa E, Pessoa... A A. Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery[J]. Optimization Letters, 2013, 7(7): 1569-1581.
- [32] Dethloff J. Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up[J]. OR-Spektrum, 2001, 23(1): 79-96.
- [33] Nagy G, Salhi S D. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries[J]. European Journal of Operational Research, 2005, 162(1): 126-141.
- [34] Brandao J C. Metaheuristics Applied to Mixed and Simultaneous Extensions of Vehicle Routing Problems with Backhauls[J]. Journal of the Operational Research Society, 2005, 56(11): 1296-1302.
- [35] Wassan N A. A Reactive Tabu Search for the Vehicle Routing Problem[J]. Journal of the Operational Research Society, 2006, 57(1): 111-116.
- [36] Avci M, Topaloglu S. A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery[J]. Expert Systems with Applications, 2016.
- [37] Jun Y, Kim B I. New best solutions to VRPSPD benchmark problems by a perturbation based algorithm[J]. Expert Systems with Applications, 2012, 39(5): 5641-5648.
- [38] Kalayci C B, Kaya C. An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery[J]. Expert Systems with Applications, 2016, 66(Dec.): 163-175.
- [39] 衷志远. 具有同时配送和收集需求的车辆路径问题研究[D]. 上海海事大学, 2007.
- [40] 谭巍, 文庆. 基于蚁群系统和 2-opt 方法求解同时送取货车辆路径 VRPSPD 问题[J]. 数学的实践与认识, 2015, 45(24): 235-242.
- [41] 曹俊阳, 杨溢, 何宇翔, et al. 考虑交通拥堵的 VRPSPD 问题及模拟退火算法优化[J]. 中国水运, 2017, 017(012): 109-111.
- [42] 周蓉, 沈维蕾. 装卸一体化车辆路径问题的自适应并行遗传算法[J]. 中国机械工程, 2018, 29(22): 2740-2749.
- [43] 盛虎宜, 刘长石, 鲁若愚. 基于共同配送策略的农村电商集送货一体化车辆路径问题[J]. 系统工程, 2019, 37(03): 102-108.

- [44] 李嘉, 杨东, 贾永基, et al. 装卸一体化电动汽车路径问题建模与优化[J]. 工业工程与管理, 2020, 25(01): 33-41.
- [45] Angelelli E, Mansini R. A branch-and-price algorithm for a simultaneous pick-up and delivery problem[J]. Quantitative approaches to distribution logistics and supply chain management, 2003: 249-267.
- [46] Mingyong L, Erbao C. An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows[J]. Engineering Applications of Artificial Intelligence, 2010, 23(2): 188-195.
- [47] Boubahri L, Addouche S-A, El Mhamedi A, et al. Multi-ant colonies algorithms for the VRPSPDTW[J]. 2011 International Conference on Communications, Computing and Control Applications, CCCA 2011, 2011.
- [48] Lai M, Tong X. A metaheuristic method for vehicle routing problem based on improved ant colony optimization and Tabu search[J]. Journal of Industrial and Management Optimization, 2017, 8(2): 469-484.
- [49] Wang H F, Chen Y Y. A genetic algorithm for the simultaneous delivery and pickup problems with time window[J]. Computers Industrial Engineering, 2012, 62(1): 84-95.
- [50] Wang C, Mu D, Zhao F, et al. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup-delivery and time windows[J]. Computers & Industrial Engineering, 2015, 83(may): 111-122.
- [51] Liu R, Xie X, Augusto V, et al. Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care[J]. European Journal of Operational Research, 2013, 230(3): 475-486.
- [52] Hof J, Schneider M. An adaptive large neighborhood search with path relinking for a class of vehicle - routing problems with simultaneous pickup and delivery[J]. Networks, 2019.
- [53] 马庆国, 孟丽君. 基于混合算法的具有硬时间窗口约束的 VRPSPD 问题[J]. 西安电子科技大学学报(社会科学版), 2009, 19(02): 41-46.
- [54] 王超, 穆东. 基于模拟退火算法求解 VRPSPDTW 问题[J]. 系统仿真学报, 2014, 26(011): 2618-2623.
- [55] 黄务兰, 张涛. 基于改进全局人工鱼群算法的 VRPSPDTW 研究[J]. 计算机工程与应用, 2016, 52(21): 21-29.
- [56] 王超, 刘超, 穆东, et al. 基于离散布谷鸟算法求解带时间窗和同时取送货的车辆路径问题[J]. 计算机集成制造系统, 2018, 24(003): 570-582.

- [57] 徐俊. 带时间窗和送取货的车辆路径规划问题研究[D]. 暨南大学, 2018.
- [58] 张庆华, 吴光谱. 带时间窗的同时取送货车辆路径问题建模及模拟求解算法[J]. 计算机应用, 2020, 040(004): 1097-1103.
- [59] 张思, 王海. 基于改进禁忌搜索算法的 VRPSPDTW 研究[J]. 物流科技, 2020, 43(07): 1-7+23.
- [60] Gillett B E, Miller L R. A Heuristic Algorithm For the Vehicle Dispatching Problem[J]. Operations Research, 1974, 22(2).
- [61] Clarke G, Wright J W. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points[J]. Operations Research, 1964, 12(4): 568-581.
- [62] Holland J. Adaptation In Natural And Artificial Systems[J]. University of Michigan Press, 1975.
- [63] Kennedy J, Eberhart R. Particle Swarm Optimization[C]. Icn95-international Conference on Neural Networks, 2002.
- [64] Glover F. Future paths for integer programming and links to artificial intelligence[J]. Computers Operations Research, 1986.
- [65] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing[J]. Readings in Computer Vision, 1987: 606-615.
- [66] Hopfield J. Neural computation of decisions in optimization problems[J]. Biological Cybernetics, 1985, 52.
- [67] Burke E, Bykov Y. The Late Acceptance HillClimbing Heuristic. Computing Science and Mathematics. 2012.
- [68] Gong W, Fialho á, Cai Z, et al. Adaptive strategy selection in differential evolution for numerical optimization: An empirical study[J]. Information Sciences, 2011, 181(24): 5364-5386.
- [69] Burke E K, Bykov Y. A late acceptance strategy in hill-climbing for examination timetabling problems. Proceedings of the Conference on the Practice and Theory of Automated Timetabling (PATAT 2008). 2008.
- [70] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by Simulated Annealing[J]. Science, 1983, 220(4598): 671-680.
- [71] Dueck G, Scheuer T. Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing[J]. Journal of Computational Physics, 1990, 90(1): 161-175.
- [72] Dueck G. New optimization heuristics: The great deluge algorithm and the record-to-

- record travel[J]. Journal of Computational Physics, 1993, 104(1): 86-92.
- [73] Appleby J S, Blake D V, Newman E A. Techniques for Producing School Timetables on a Computer and their Application to other Scheduling Problems[J]. Comput. J., 1961, 3(4): 237-245.
- [74] Liu F, Gui M, Yi C, et al. A Fast Decomposition and Reconstruction Framework for the Pickup and Delivery Problem with Time Windows and LIFO Loading[J]. IEEE Access, 2019, PP(99): 1-1.
- [75] Glover F: NEW EJECTION CHAIN AND ALTERNATING PATH METHODS FOR TRAVELING SALESMAN PROBLEMS[N], Computer Science and Operations Research, 1992.
- [76] Hansen P, Mladenovi N. Variable neighborhood search: Principles and applications[J]. European Journal of Operational Research, 2001, 130(3): 449-467.
- [77] Silva M L, Souza S D, Souza M F, et al. A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems[J]. Expert Systems with Applications, 2019, 131(OCT.): 148-171.
- [78] Robbins H. Some aspects of the sequential design of experiments[J]. Bulletin of the American Mathematical Society, 1985, 58(5): 527-535.
- [79] Castro, Olacir, R., et al. Evaluating selection methods on hyper-heuristic multi-objective particle swarm optimization[J]. JOURNAL OF HEURISTICS, 2018, 24(4): 581-616.
- [80] Maashi M, Özcan E, Kendall G. A multi-objective hyper-heuristic based on choice function[J]. Expert Systems with Applications, 2014, 41(9): 4475 - 4493.
- [81] Sabar N R, Ayob M, Kendall G, et al. A Dynamic Multiarmed Bandit-Gene Expression Programming Hyper-Heuristic for Combinatorial Optimization Problems[J]. IEEE Transactions on Cybernetics, 2015, 45(2): 217-228.
- [82] Auer P. Finite-time analysis of the multiarmed bandit problem[J]. Machine Learning, 2002, 47.

攻读硕士学位期间取得的研究成果

一、已发表（包括已接受待发表）的论文，以及已投稿、或已成文打算投稿、或拟成文投稿的论文情况（只填写与学位论文内容相关的部分）：

序号	作者（全体作者，按顺序排列）	题 目	发表或投稿刊物名称、级别	发表的卷期、年月、页码	相当于学位论文的哪一部分（章、节）	被索引收录情况
1	刘发贵，桂梦珂，易辰，兰宇琳	A Fast Decomposition and Reconstruction Framework for the Pickup and Delivery Problem with Time Windows and LIFO Loading	IEEE Access (JCR -Q1)	已发表 Volume:7 2019 年 6 月， pp:71813-71826	第四章	SCI(WOS:000471975900001)
2	桂梦珂，刘发贵，兰宇琳，赖承启	A Hybrid Heuristic Algorithm for Urban Distribution with Simultaneous Pickup-delivery and Time Window	European Journal of Operational Research (JCR -Q1)	已投稿	第三章 第四章	SCI
3	兰宇琳，刘发贵，吴永贤，桂梦珂	Decomposition Based Multi-Objective Variable Neighborhood Descent Algorithm for Logistics Dispatching	IEEE Transactions on Emerging Topics in Computational Intelligence (JCR -Q1)	已发表 DOI:10.1109/TETCI.2020.3002228		SCI

二、与学位内容相关的其它成果（包括专利、著作、获奖项目等）

发明专利

- 1) 获取带时间窗和先进后出限制的取送货车车辆路径的方法（实审阶段）

公开号：110097313A

发明人：刘发贵，易辰，桂梦珂

- 2) 一种考虑逆向物流的同城配送路线规划方法（预审阶段）

发明人：刘发贵，桂梦珂

参与科研课题：

- 1) 一站智能供应链服务平台。2020 年广州市工信局项目，项目编号：x2jsD8202010。

2019.12-2022.12.

- 2) iTMS 智能供应链管理系统。2018 年广州市“中国制造 2025”产业发展资金项目。项目编号：x2jsD8183470。2018-2021。

3) 广东省物流供应链与物联网工程技术研究中心。广东省工程技术研究中心建设项目，粤科产学研字〔2016〕176号。2016.3-2019.3。

致 谢


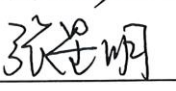


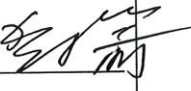

时光荏苒，三年的研究生生涯已经接近尾声，回顾这三年的科研生活，有压力山大，有欢声笑语，还有新冠肺炎。令人尊敬的老师们和并肩作战的同学们给我带来了十分美好的回忆，让我在学业方面、思维方面还有为人处世方面都得到了锻炼和提升。在此，谨向所有帮助我的老师、同学和家人朋友们表达我衷心的感谢。

首先，我要向我的导师刘发贵教授致以诚挚的感谢，不仅在研究课题的选择和论文的撰写上给予我很大的指导，平时也乐于帮我解答任何学术上的苦恼和困惑。在我遇到难以克服的挑战时，刘老师也能及时开导和鼓励，让我信心倍增。在我懈怠时，能及时督促我，让我摆正学习态度。更令人敬佩的是她一丝不苟的生活作风和持续学习的科研精神，是我永远学习的榜样。再次感谢刘老师的谆谆教诲，我会将这些教诲铭记于心，在以后的工作中继续努力称为更优秀的人！

其次，我要感谢实验室的兄弟姐妹们，给我带来了许多欢乐。特别感谢易辰师兄、兰宇琳师姐、杨鑫师兄和赖承启师弟，在我的科研过程中给予了我非常大的帮助，积极地与我交流约束问题，耐心的帮我答疑解惑；感谢同届的谷典、郑来磊、马振江和卢云圣这几位同学，大家相互学习和分享，共同进步！

感谢华南理工大学对我的培养，为我提供了生活和学习的环境。感谢计算机科学与工程学院所有任课老师的教导，让我收益良多，辛苦了！

最后，感谢我的家人对我的支持和付出，让我增长了见识，拓宽了视野，给予我前行的勇气和动力！

<p>3. 答辩委员会对论文的评语</p> <p>(主要包括: 1.对论文的综合评价; 2.对论文主要工作和创造性成果的简要介绍; 3.对作者掌握基础理论、专业知识程度、独立从事科研工作能力以及在答辩中表现的评价; 4.存在的不足之处和建议; 5.答辩委员会结论意见等)</p> <p>桂梦珂同学的学位论文从物流领域中降低运输成本的难题出发, 针对带时间窗和同时取送货的车辆路径问题展开研究工作, 选题具有一定的理论和实践价值。</p> <p>在充分调研目前的主流算法和国内外研究现状的情况下, 针对目前存在的问题和缺陷, 论文设计并实现了具有创新性的新兴混合式智能启发式算法, 加强了基础智能启发式算法的后期搜索能力。针对局部搜索策略中关于搜索深度和搜索广度之间的平衡问题, 论文设计了一套多臂老虎机智能选择算法。基于标准数据集和真实运输数据集的实验研究验证了所提算法的有效性。</p> <p>论文研究方法合理, 实验较充分, 反映了答辩人具有扎实的专业理论基础, 对所研究问题有较为全面的了解。作者已发表一篇 SCI 源刊文章, 申请发明专利一项, 达到了华南理工大学硕士学位论文的要求。</p> <p>在答辩过程中, 答辩人表述清楚, 能准确理解并回答专家提出的问题。经答辩委员会无记名投票表决, 一致同意通过桂梦珂同学的硕士学位论文答辩, 建议授予硕士学位。</p>	
<p>论文答辩日期: 2021 年 06 月 04 日 答辩委员会委员 5 人</p> <p>表决票数: 同意毕业及授予学位 (5) 票</p> <p> 同意毕业, 但不同意授予学位 () 票</p> <p> 不同意毕业 () 票</p> <p>表决结果 (打“√”): 同意毕业及授予学位 (√)</p> <p> 同意毕业, 但不同意授予学位 ()</p> <p> 不同意毕业 ()</p>	
<p>答辩委员会 成员签名</p>	<p> (主席)</p> <p>   </p>
<p>答辩秘书 签名</p>	<p></p>