

摘 要

配送是物流系统中很重要的一个环节，是一系列狭义的物流活动的集成，它要求在规定的时间内以一定的方式将确定的货物送到指定的地点。而车辆路径问题是研究货物运输成本最小的物流配送问题。车辆路径问题是运输组织优化中的核心问题，由于它将运筹学理论与生产实践紧密地结合，因此近几十年取得了丰富的研究成果，并且被称为“最近几十年运筹学领域最成功的研究之一”。

本文分析和总结了车辆路径问题的历史和研究现状，以及常用模型、时间复杂度，以综合性能的角度对求解 VRPTW 问题的算法进行了一定的归纳和分析，并在此基础上确定了本文的研究方法和目标。同时结合实际提出了本文的研究问题——带时间窗的车辆路径问题(VRPTW)，建立了以车辆容量和客户需求等为约束条件，以配送运输成本为目标函数的 VRPTW 数学模型。提出了一种两步优化的实现策略。第一步，选取动态的 push forward insertion heuristics (Stochastic PFIH) 算法产生机制构建问题的初始解，保证了初始解的多样性，同时在产生的初始解中设置了限制条件，实现对初始解的筛选，为第二阶段路径优化提供了高质量的初始解；以改进的大规模邻域搜索为新解产生机制，并将模拟退火算法和大规模邻域搜索算法混合来进行路径优化，得到问题的最优解，该混合算法充分利用了两种算法的优点，克服了它们的缺点。第二步，提出一种时间窗修正规则来调整时间窗，使得车辆在每个客户的等待时间为零，真正实现了路径的优化，节省了成本，并给出了理论证明。最后通过 VC++编程在计算机上实现，以 Solomon 的标准数据中的 C101 系列数据进行数值实验，实验结果验证了本文算法的有效性。

关键词：车辆路径问题，模拟退火算法，大规模邻域搜索，回归迭代策略

Abstract

Distribution plays an important role in logistic system, it is the integration of a series of logistic campaign in a narrow sense, and logistic distribution requires delivering the right commodities to the right place by the right method at the right time. Vehicle routing problem involves the design of a set of minimum-cost vehicle routes, delivering the right commodities to the right location by the very method at the very time. So vehicle routing problem represents a very important part of any logistic distribution systems and they are named as one of the most successful areas in Operations Research in the past decades.

In this dissertation, the main research work and innovative points as follows: We firstly describe the characteristics and classification of these problems, and the state-of-the-art in the solution algorithms. On the base of reality, we purpose the research problem ---vehicle routing problem with time window (VRPTW) and construct a mathematical model: defines an objective function. Meanwhile, we give a review of the past research on vehicle routing problems and their solution methods, establishing the base of algorithm in this dissertation. In this paper, we propose a two-stage optimization strategy for VRPTW. In the first step, for constructing a good initial solution, this work used the stochastic PFIH, which based on the algorithm known as Push-Forward Insertion Heuristic (PFIH), guaranteeing the diversity of the initial solution, at the same time, give a limit in the process of the constructing initial solution, realizing the filtration of initial solution. The producing mechanism of the new solution is betterment of large neighbor search, then a different hybrid system based on the combination of SA and LNS is proposed to optimize the initial solution. In the second step, a regression iterative strategy is put forward to tune time windows for the customers and figure out the best time for each vehicle departing, it can make the total waiting time zero. Finally, this method is implemented on computer in VC++. It is proved by the experiment that the hybrid strategy can solve VRPTW efficiently and quickly. Comparing with other algorithms, it has practicality and effectiveness, simultaneously, author provide an efficient algorithm to solve VRPTW.

Key words: simulated annealing, large neighbor search, vehicle routing problem with time windows, regression iterative strategy.

独创性声明

本人声明,所提交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知,除了文中特别加以标注和致谢的地方外,论文中不包含其他人已经发表或撰写过的研究成果,也不包含为获得武汉理工大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签名: 蒋文霞 日期: 2007年11月29日

关于论文使用授权的说明

本人完全了解武汉理工大学有关保留、使用学位论文的规定,即学校有权保留、送交论文的复印件,允许论文被查阅和借阅;学校可以公布论文的全部或部分内容,可以采用影印、缩印或其他复制手段保存论文。

(保密的论文在解密后应遵守此规定)

签名: 蒋文霞 导师签名: 彭林山 日期: 2007年11月29日

第 1 章 绪论

1.1 课题研究背景与意义

随着市场竞争的日益加剧、世界经济一体化进程的加快和科学技术的飞速发展,许多企业将被称为“第三方利润来源”的物流作为提高市场竞争能力和提升核心竞争能力的重要手段,将先进的物流理论和物流技术引入企业的生产和经营管理中。

配送是物流系统中很重要的一个环节,是一系列狭义的物流活动的集成,以送货上门为目的,有确定组织、为实现配送成本的降低,必须对配送运输进行合理规划。配送运输的合理规划涉及到时间、成本、环境三方面的因素,首先从时间上要考虑准时性、快速响应;成本上要考虑配送运输涉及的各种开支(车辆的购置成本和损耗、司机薪酬等等);环境上要尽可能减少不必要的行驶,避免造成交通拥挤、空气以及噪音等污染。这些可以通过改善运输方式、路径规划来加以改进。

配送运输的路径规划直接或间接影响时间、成本和环境三方面因素,配送路线越长,占用的车辆越多,配送的成本也越高。从理论上讲,配送的路线规划问题是组合优化问题的一种,对应的优化目标是在满足车辆容量等约束条件的前提下,使运送货物的车辆数、时间、路程的组合达到最小。

物流配送要求在规定的时间内以正确的方式将需求的货物送到指定的地方。而车辆路线问题则研究总成本最小的车辆路线,在合适的时间以合适的方式将正确的货物运送到正确的地点。据文献记载,1997年,美国的超额(excess)运费高达450亿美元,在俄罗斯、法国和丹麦的运输成本分别占了全国总支出的15%、9%和15%。同年,食品行业的配送成本占总物流成本近一半,而附加值商品的配送成本占用比例更是高达70%。据1989年Larsen的一篇报告,76.5%的货物运输是通过车辆来完成的。这显示了车辆路线问题在物流配送中有着广泛的应用背景^[1]。

由于车辆路径问题将运筹学理论与生产实践紧密地结合在一起,近几十年取得了很多成果,因此被称为“最近几十年运筹学领域最成功的研究之一”。许多学者认为VRP的成功归因于理论和实践的紧密联系:具有学术背景的运筹学学

者不仅设计和改进了形式多样的模型和算法，而且对推动路径系统在实际中的应用也起到重要作用；另一方面，由于符合生产需要的计算机软件的成功研制，工商业者对车辆路径问题的重视也日益加强。

然而，长期以来，由于缺乏 GIS/GPS 技术，车辆路线问题的研究主要是理论模型的算法研究。现今，随着 GIS/GPS 技术的广泛应用，对车辆路线问题展开面向应用的研究(包括算法研究)将具有现实意义。

1.2 问题的提出

车辆路径问题中最常见的决策问题就是，寻找合适的配送工具在最佳的路线上进行配送，同时要使总配送时间或配送距离尽量短，从而降低配送成本、改善服务水平，同时降低企业运作成本，提高企业经济效益。

而对现实问题进行理论研究，需要我们抽象出一个既便于计算机处理又尽量真实反映实际问题的模型。理论上，车辆路径问题可以追溯到旅行商^[2](Traveling Salesman Problem, TSP)，该问题研究的是一个旅行商从城市网络中的某节点出发，遍历城市网络所有节点而且访问每个节点仅仅一次后回到起始点的行程路线安排。TSP 只包括了地理因素，即只考虑了地理上的路径最短问题。而在实际问题中，往往还要考虑各种约束条件，比如车辆的载重，行使所需时间，顾客需求等等。

车辆路径问题^[3](Vehicle Routing Problem, 简称 VRP)是一种特殊的 TSP 问题，其中配送网络中每个节点都存在不同的货物需求，负责配送的车辆有容量限制。因此 VRP 不像 TSP 那样只单纯考虑地理上的最短路径，还要考虑各节点的顾客需求量以及车辆容量限制等实际的约束条件，比较而言，对 VRP 的研究更具有实际的应用价值。

VRP 是许多实际路径规划问题的基本模型，实际的运输路径规划往往根据需要给问题加上一些有意义的约束，形成了各种 VRP 的变形问题。在各种车辆路线问题中，一个基本问题是带时间窗口的车辆路线问题(VRPTW)。其主要内容是：以成本最小的目标安排多辆车有序地前往配量给定的各配送点而构成的配送路线，其中每辆车必须从同一车站出发并最后返回车站，每个配送点只能安排一个车次在限定的时间窗内配送，且每一条配送路线不得超过车辆的装载容量和车辆的最后返回时间：如果车辆提前到达配送点，则需要等待，直到在

时间窗内才能配送。

由于 VRPTW 是 NP-hard^[4]问题,因而目前对其研究大多集中在求解最优或近似最优解的算法的发现和改进上,本文基于这种目的,着重于对 VRPTW 的混合算法进行研究。

1.3 国内外的研究现状

车辆路径问题最早是由 Dantzig 和 Ramser 于上个世纪 50 年代末期在一篇学术论文中提出的。当时的车辆路径问题主要集中在静态的车辆调度问题上,描述的是运筹学中的优化问题。在国外,对 VRP 的理论研究和实际应用都已经取得了非常显著的成果。随着研究的深入发展,如何使研究的理论模型更贴近现实中的路径规划问题已成为研究者们关注的焦点。由于其现实意义,近几十年来已经出现了许多关于带时间窗的车辆路径问题的研究文献。

最早研究带时间约束的路线问题是 1981 年 Christofides 在一篇技术报告中对带时间约束的旅行售货员问题(TSPTW)的优化算法研究。而最早发表的研究文献则是 1983 年 Baker 对 TSPTW 问题提出一种最优化算法^[8]。最早对 VRPTW 问题的算法研究是 1986 年 Solomon^[9]的启发式算法研究,1987 年 Kolen 等^[10]最早提出最优化算法:动态规划算法,这是第一个 VRPTW 最优化算法,该算法通过搜索树实施分支,应用动态规划求解带时间窗约束的最短路问题(SSPTW)确定下界。1987 年 Solomon 较系统地将 VRP 问题的启发式方法推广应用到 VRPTW 问题^[11],通过最坏情形的分析指出求解 VRPTW 问题从根本上比求解 VRP 问题更难。1992 年,Desrochers 等结合线性规划松弛和列生成法提出一种 VRPTW 问题的最优化算法^[12],成功求解了一些 100 个点规模的问题。1994 年, Fisher 应用最小化 K-树方法提出 VRP 问题的一种最优化算法,并推广到求解 VRPTW 问题^[13],同时指出这类算法适用于求解时间窗口比较宽松的问题,还通过计算试验验证了这一点。1997 年 Kohl^[14]通过 Lagrangian 松弛求出 VRPTW 的下界后,应用分支界定法对 VRPTW 进行优化求解,并成功求解了规模为 100 各点的问题。

由于 VRPTW 是 NP-hard 的问题,这意味着所有能精确求解这类问题的算法在最坏的情况下需要指数级的时间。采用精确算法虽然可以得到最优解,但庞大的计算量会消耗大量的计算时间,只能用于解决有限规模的问题,不适用于

现实中的大规模配送路径问题，已往的研究结果也证明了这一点。

在上面提到的这些方法的基础上，后来的研究者提出了各种改进解的方法求解 VRPTW^{[17][18]}，通过初始解中边的交换和结点的交换来改进初始解的性能。由于启发式方法在求解大规模复杂问题方面具有优越性，国外许多研究者都致力于研究新的高效率的启发式算法来构造或改进 VRPTW 的解。近几年来禁忌搜索法(Tabu Search, 简称 TS)、模拟退火(Simulated Annealing, 简称 SA)以及遗传算法(Genetic Algorithms, 简称 GA)等亚启发式方法陆续被引入到 VRPTW 的求解中^[20]，并取得了显著优于传统启发式方法的结果。蚁群算法(Ant Colony System, 简称 ACS)作为一种比较新的启发式仿生算法，也被应用于 VRPTW 问题求解中^[20]，研究结果表明该算法具有良好的性能。

由于这些启发式方法具有强大的全局优化性能和通用性，因此将这些算法进行混合来求解 VRPTW 已成为目前研究者们关注的一个领域。如 Hon Wai Leong 和 Ming Liu^[51]在路线构造阶段采用贪婪算法 PFIH, 在路线优化阶段采用可以进行信息交流的多级方式来执行全局优化。Alvarenga G. B 和 Mateus G.^[53]采用动态 PFIH 构造初始解，结合快速遗传算法和 SPP(Set Partitioning Problem)构成的 CGH (Column Generation Heuristic) 算法。Andrew Lim ,Fan Wang^[53] 采用 GRASP 反复使用局部搜索的结果作为下一代中贪婪算法结构的输入。第一阶段采用随机的贪婪算法产生初始解，第二阶段通过动态禁忌算法进行局部搜索。Andrew Lim 等^[54] 结合 SWO 和 PFIH 构造第一阶段的初始解，第二阶段采用局部搜索，和一些经典算子 E-opt (enumeration-based operator) GEC(Generalized Ejection chain-based operator)来减少总距离。Oliveira, H. C. B.等^[55] 第一阶段由动态 PFIH (任意从一个客户开始产生每条路线)产生初始解，第二阶段用 SA 进行局部搜索，并采用爬山操作构成混合算法，以此优化解。

在我国，一些专家和学者也开始尝试 VRPTW 的研究。1998 年，李大卫等人^[56]对适用于 TSP 的最近距离搜索启发式算法进行修正，构造出评价函数，并据此提出一个求解 VRPTW 的启发式算法。2000 年，谢秉磊等人^[57]将货运量约束和时间窗约束转化为目标约束，设计了基于自然数编码的可同时处理软、硬时间窗约束的遗传算法，实验分析获得了较好的结果。2003 年，宋厚冰和蔡远利^[58]针对 VRPTW 问题，在标准遗传算法的基础上，将分组信息与每一个染色体结合，并辅以补交换局部搜索技术，构造了一种改进遗传算法，使得求解结果更接近最优解。2004 年，刘小兰等人^[59]为了克服原有大规模邻域搜索算法不

能有效求解时间窗较宽 VRP 的缺陷,介绍了 VRPTW 的通用数学模型。通过分析各主要变量之间的关系,构造了一种简单、快速的确定性初始算法,并通过引入“短路径优先策略”,构造了改进的大规模邻域搜索算法,该策略也可嵌入到求解时间窗比较窄的 VRP 中,达到加速搜索的目的。吴璟莉、李陶深^[60]探讨了如何将基于遗传算法和禁忌搜索算法的混合策略。王德东^[61]等人提出了一种新的 GSAG 算法,该算法包含了遗传算法和模拟退火算法的优点。

1.4 本文研究内容

本文首先对车辆路径问题按照涉及到的实际配送路径规划中的不同约束条件进行分析和总结,确定本文的研究对象为带时间窗的车辆路径问题(VRPTW),即在不违背车辆容量限制、满足客户时间窗口等约束条件下,合理规划配送时的车辆路径安排,以尽可能小的成本满足客户对货物和服务时间区间的要求;然后从解决方法的角度对求解车辆路径问题的各种算法进行了分类和总结,从而奠定了本文算法的基础;接着根据结合实际提出了本文的研究问题——带时间窗的车辆路径问题(VRPTW)。建立了以车辆容量和客户需求等为约束条件,以配送运输成本为目标函数的 VRPTW 数学模型。提出了一种两步优化的实现策略。第一步,选取动态的 push forward insertion heuristics (PFIH) 算法产生机制产生初始解;以大规模邻域搜索为新解产生机制,并对该方法进行了改进;最后,提出了一种混合模拟退火算法和大规模邻域搜索算法的混合算法。第二步,提出一种时间窗修正规则来调整时间窗,使得车辆在各个节点的等待时间为零,并给出了理论证明。最后通过 VC++ 编程在计算机上实现,以 Solomon 的标准数据中的 C101 系列数据进行数值实验,实验结果验证了本文算法的有效性。

本文的结构安排如下:

第一章 绪论:介绍本文的研究背景与意义,并提出本文的研究目标、研究思路及研究框架。

第二章 描述和定义一般车辆路径问题,并针对实现中的约束条件,对车辆路径问题的原型和各种衍生问题进行了简明的综述。重点对带时间窗的车辆路径问题进行了理论上的描述,并建立了 VRPTW 的数学模型,为后文的研究奠定了基础。

第三章 在分析 VRPTW 问题的复杂性的基础上，对带时间窗的车辆路径问题算法进行综述，为本文的算法提供了依据。

第四章 本章首先介绍了初始解的生成方法，并阐述生成初始解的过程，介绍了模拟退火算法和大规模邻域算法，在此基础上提出本文的混合算法，同时介绍了本文提出的回归迭代策略，这部分是本文的重点。

第五章 用本文提出的混合算法求解 VRPTW 问题，并用 Solomon 的标准测试问题中的 C101 类进行了实验，与当前最优解进行了简单的比较。

第六章 总结与展望：对本文进行归纳，概括了本文的主要工作和创新点，提出下一步研究工作和研究方向。

第 2 章 车辆路径问题

车辆路径问题 (Vehicle Routing Problem, VRP) 最早由著名学者 Dantzig 和 Ramser 于 1959 年提出^[5], 而后由 N.Christofides 对其进行总结深化^[6]。由于该问题将运筹学理论研究和实际生产生活紧密联系起来, 因此提出后即显现出旺盛的生命力。到目前为止, 该问题的应用不仅仅局限在交通运输领域, 还被拓展到电路设计、电网布局、转炉计划、铁路运输、公交调度、水道航线等领域。可以说, VRP 是运筹学领域研究最活跃、成果最精彩的方向之一。

经过几十年的研究, VRP 领域取得了相当丰富的成果。为了把该问题的研究现状清晰的呈现在众人面前, 本章对该问题的描述和研究成果做了以下回顾和总结, 然后详细描述了本文重点研究的带时间窗的车辆路径问题 (VRPTW)。

2.1 旅行商问题

最著名同时也是最简单的车辆路径问题是旅行商问题 (Traveling Salesman Problem, TSP), 旅行商问题是 VRP 问题的一个特例 (当 VRP 问题中只包括一条路径, 且没有容量约束时, 就成为旅行商问题), TSP 问题是运筹学、图论及组合优化中的著名难题, 由于其有广泛的应用背景, 引起了人们极大的兴趣, 现已归入 NP-hard 问题类。

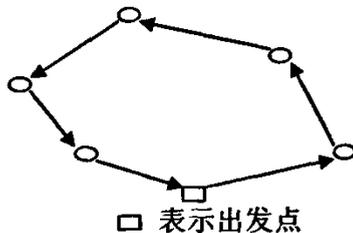


图 2-1 一个典型的 TSP 问题的解

TSP 的一般描述为: 旅行商要到若干城市, 从驻地出发, 经过其余各城市一次且仅仅一次, 然后返回到出发点, 求其最短行程, 即寻找一条旅行路线, 使总的旅行距离 (时间或费用等) 最少。一个典型的 TSP 问题的解如图 2-1 所示。实际中有很多应用问题都可以归结或转化为旅行商问题, 如物质运输路线问题、管道铺设问题等等, 同时它又是许多复杂模型的基础。

其数学模型描述如下^[1]:

$$\text{Min} \sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ij} \quad (2-1)$$

$$\sum_{i=1}^N x_{ij} = 1 \quad (j = 1, \dots, N) \quad (2-2)$$

$$\sum_{j=1}^N x_{ij} = 1 \quad (i = 1, \dots, N) \quad (2-3)$$

$$x_{ij} = 0, 1 \quad (i, j = 1, \dots, N) \quad (2-4)$$

其中 $x_{ij} = \begin{cases} 1, & \text{若路线中含弧}(i, j) \\ 0, & \text{否则} \end{cases}$, c_{ij} 表示点 i 与 j 之间的成本 (可以是路长或

运行时间), N 表示节点数目。

式 (2-1) 为目标函数, 式 (2-2)、(2-3)、(2-4) 为问题的约束条件。

2. 2 车辆路径问题的数学模型分析

如果在 TSP 中将售货员对应一辆车, 售货员的出发点对应车辆的车站, 其它点对应需要供货或卸货的点, 再对每个路线施加车辆装载容量限制, 则成为著名的 VRP 问题 (Vehicle Routing Problem)。

一般情况下, VRP 的描述是: 对一系列给定的客户 (送货点或取货点) 确定适当的配送车辆行驶路线, 使其从配送中心出发, 有序地通过它们, 最后返回配送中心, 并在满足一定的约束条件下 (如车辆载重量、客户需求量、时间窗限制等), 使总运输成本最小 (如使用车辆数最少、车辆行驶路程或时间最短等)。总结起来, VRP 问题是以成本最小为目标, 寻求满足以下约束的解:

- 1) 每辆车以同一中心库为起点和终点。
- 2) 每个需求点必须由一辆车完成其供货或卸货。
- 3) 每辆车的总装货量不得超过车辆的装载容量。

VRP 的数学模型如下^[1]:

$$\text{Min} \sum_{i \in N} \sum_{j \in N} c_{ij} \sum_{k \in K} x_{ijk} \quad (2-5)$$

$$s.t. \sum_{i \in N} \sum_{k \in K} x_{ijk} = 1, j \in N \quad (2-6)$$

$$\sum_{j \in N_0} \sum_{k \in V} x_{ijk} = 1, i \in N \quad (2-7)$$

$$\sum_{i \in N_0} x_{ipt} - \sum_{j \in N_0} x_{ptk} = 0, p \in N_0, k \in V \quad (2-8)$$

$$\sum_{i \in N} d_i (\sum_{j \in N_0} x_{ijk}) \leq Q, k \in V \quad (2-9)$$

$$\sum_{j \in N} x_{0jk} \leq 1, k \in V \quad (2-10)$$

$$\sum_{i \in N} x_{i0k} \leq 1, k \in V \quad (2-11)$$

$$X \in S, X = (x_{ij}) = (\sum_{k \in V} x_{ijk}) \quad (2-12)$$

$$x_{ijk} = 0, 1, i \in N_0, j \in N_0, k \in V \quad (2-13)$$

其中符号说明如下：

N ：配送需求点下标集合，即 $N = \{1, 2, \dots, n\}$ ，其中 n 为配送需求点数。

点 0：表示车站，并记 $N_0 = N \cup \{0\}$ 。

V ：车辆下标集合，即 $V = \{1, 2, \dots, K\}$ ，其中 K 为车辆数。

Q ：每辆车的装载容量（所有车辆的装载容量相同）。

d_i ：点 i 的配送需求量，假设 $\max_{i \in N} d_i \leq Q$ 。

c_{ij} ：从点 i 直接到达点 j 的运输成本。

$$x_{ijk} = \begin{cases} 1, & \text{若车辆 } k \text{ 经过边 } (i, j) \\ 0, & \text{否则} \end{cases}$$

式子 (2-5) 表示目标函数，约束 (2-6) 和 (2-7) 确保每个配送需求点恰好被车辆访问一次。(2-8) 是流守恒条件，即车辆到达某点后，必须离开该点。

(2-9) 是车辆容量限制。约束 (2-10) 和 (2-11) 限制车辆最多只能有一条回路。

(2-12) 限制在车辆路线中不含子回路。由于通过 (2-6) 和 (2-8) 可推出 (2-11) 成立，因此约束 (2-10) 是多余约束，可以去掉。类似地，通过 (2-8) 和 (2-11) 可推出 (2-12) 成立，也可以去掉约束 (2-12)。如果没有约束 (2-9)，则 VRP 退化为 MTSP 问题。注意到 $\sum_{j \in N} x_{0jk}$ 的值表示是否启用了车辆 k ，因此，可以在目

标函数中加上车辆启用成本 $f_k \sum_{k \in V} x_{0k}$ 来减少车辆数。图 2.2 显示了 VRP 的一个解。

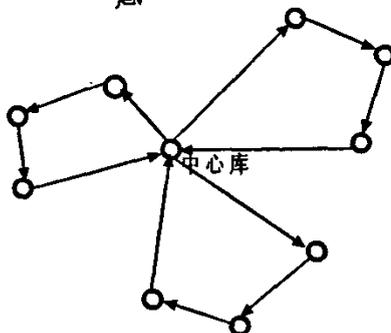


图 2-2 VRP 的一个解

2.3 车辆路径问题的分类

从以上对 VRP 问题的描述可以看出此类问题由几个关键的要素组成：目标、配送中心、车辆、客户，而这几个要素的变化或组合又可以引申出 VRP 问题的一些变形，这些变化又来自于实际应用，相应地其模型构造及算法设计也就差别很大。现将车辆路径问题的类型归纳如下^[62]。

按已知信息的特征分为确定性 VRP 和不确定性 VRP，其中不确定性 VRP 又可进一步分为随机车辆路径问题 (SVRP) 和模糊车辆路径问题 (FVRP)，其中 SVRP 又可分为客户随机 VRPSD 和需求量随机 VRPSC 以及两者的结合 VRPSCD。

按约束条件可分为带有容量限制的车辆路径问题 (CVRP)、带时间距离约束的车辆路径问题 (DVRP) 以及带有时间窗的车辆路径问题 (VRPTW)。

按需求是否可切分，又可分为可切分的 VRP 和不可切分的 VRP；按顾客需求量是否超过车的容量来分，可分为满载车辆路径问题和非满载车辆路径问题。

按配送中心的多少可分为单车场车辆路径问题 (SVRP, 等同于一般的 VRP)、多车场车辆路径问题 (MVRP)，其中，MVRP 又可以根据是否每辆车都有固定的终点车场分为终点车场固定的车辆路径问题和终点车场不固定的车辆路径问题。

还有带回程运输的 VRP (VRP with Backhauls, VRPB) 和带取送货的 VRP (VRP with Pickup and Delivery, VRPPD) 等等。

对于车辆路径问题的研究，考虑的约束条件越丰富越细致，往往越接近于实

际运输问题，但是解决问题的难度也随之大大增加。而事实上，有些复杂的问题可以转化为简单问题的组合，这样就可以降低难度。本文将带有时间窗的车辆路径问题作为主要研究对象，将在下一节详细介绍。

2.4 有时间窗车辆路径问题 (VRPTW) 分类

如果在 VRP 中增加了配送需求点的时间窗约束，则成为带时间窗的车辆路径问题 (Vehicle Routing Problem With Time Windows, VRPTW)。下面几节的内容是此类问题的详细描述。

2.4.1 VRPTW 的概念及基本描述

在 VRP 问题的基础上，给每个客户节点 i 加上接受服务所允许的时间窗口约束，即 $[e_i, l_i]$ ，这样 VRP 问题就扩展为了 VRPTW 问题。

VRPTW 问题可描述如下：在不违背车辆限制的条件下，用于送货的若干车辆从配送中心出发，在返回配送中心之前，应使总成本（路程或时间等）最低，同时还要满足各客户的需求（在指定时间窗内服务客户）、不能使任何车辆超载。

由于现实生活中，许多与时间相关的运输调度问题都可以归结为 VRPTW 问题来处理，如邮局邮件的投递、电子商务环境下的网上销售货物的配送、学校校车的安排等。鉴于其现实意义，近年来许多领域的学者对带时间窗的车辆路径问题进行了广泛深入的研究。

2.4.2 VRPTW 时间窗的分类

由于 VRPTW 问题的复杂性^[62]，如果严格按照顾客设定的服务时间为其服务，可能造成企业的配送成本增加。如果允许在某些客户点适当地延误，可能使运输成本大为减少，即对该延误现象给予一定的惩罚。所以根据决策者在对客户满意和成本二者权衡时偏好的不同时，若按照客户满意度来分，时间窗还可以分为硬时间窗 (Hard Time Windows, 简称 HTW) 和软时间窗 (Soft Time Windows, 简称 STW) 以及混合时间窗 (Mixed Time Windows, 简称 MTW) 三种情况。

硬时间窗(Hard Time Windows): 指配送车辆必须在规定时间段内将配送货

物送达到顾客手中，顾客拒绝接受在此时间段之外提供的服务。如图 2-3 为一惩罚函数(Penalty Function)，当配送货物送达时间超过了规定的时段(e, l)，其惩罚值 $P(t)$ 等于一个非常大的正值，以表示硬时间窗的限制。

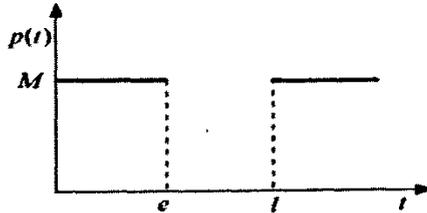


图 2-3 硬时间窗

硬时间窗的车辆路径问题(VRPHTW)中，车辆可以在最早服务时间之前到达客户所在地，但必须等待直至到了最早服务时间才能对客户提供服务；另外车辆不能在最晚服务时间之后到达客户所在地。如在 Just-In-Time 生产系统中，送货车辆到达时间若迟于指定的最晚到达时间，会导致整个生产线的延误和闲置，造成时间成本的增加和效率的降低，因此这种情况下往往不允许违背时间窗的约束。对于 VRPHTW 问题来说，服务所有客户所需的车辆数目也是决策变量，一般来说，所用车辆越少，对应的成本也越小，如何用最少的车辆在不违背车辆容量及客户时间窗的约束的前提下，为所有客户提供服务，也是决策者关心的一个问题。因此在对 VRPHTW 建模时确定的目标函数往往是多目标且多层次的，根据具体要求的不同，目标函数的层次也不尽相同。通常，首先最小化服务所有客户所需的车辆数目，因为每多增加一辆车意味着多增加一条服务路线和一名司机，其成本会高很多，而且需要的车辆越多，车队的购置成本也越高；其次，对于使用同样数目车辆的情况，最小化总的行驶路程或行驶时间。

软时间窗(Soft Time Windows)：指配送车辆如果无法将货物在特定的时段(如图 2-4 中的(e, l))内送到顾客手中，则必须按照违反时间的长短施以一定的罚金或其它惩罚法则。图 2-4 就是一种可能的惩罚函数。

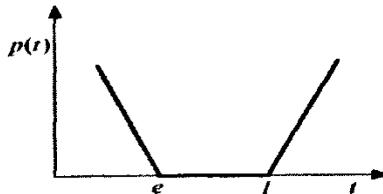


图 2-4 软时间窗

带软时间窗的车辆路径问题(VRPSTW)相对 VRPHTW 来说放松了对时间窗

的约束，这是因为在实际情况下，由于道路交通流量，车辆的行驶速度以及客户的需求时间等不确定因素导致货物无法在规定的时间内送到，如果采用硬时间窗的约束进行优化，将导致成本非常之高，甚至可能无解。而 VRPSTW 只要求车辆应尽可能在客户指定的时间窗内以最小的总成本为所有客户提供服务，允许一定程度的延误现象。显然，VRPSTW 较 VRPHTW 具有更好的通用性。在实际运输规划中，决策者往往需要在客户满意和成本二者之间权衡，偏好不同决定了可以允许延误的程度不同。在 VRPSTW 中，车辆如果在最早服务时间之前到达客户所在地，需等待直至到了最早服务时间后才能提供服务；如果在最晚服务时间之后到达客户所在地，则可能导致客户满意度降低，隐性成本的增加。因此应尽量避免延误现象的发生，如果发生，应加以惩罚，惩罚的具体程度可由决策者设定。在对 VRPSTW 建模时，表现为目标函数增加了一项惩罚项，它具体体现在目标函数值(对应服务成本)的增加。

混合型时间窗(Mixed Time Windows): 在系统中有些顾客属于硬时间窗，有些属于软时间窗；对同一顾客，又可能软、硬时间窗混合使用。在实际的 P(t) 配送工作中，车辆如果能在最佳时间段(如图 2-5 中 (e, l))内将货物送达到顾客手中，则不处罚；若在图 2-5 中的 $[(a, e)$ 或 $(l, b)]$ 时段内才送达，则顾客的满意度降低，将处以一定的惩罚(转化为惩罚函数)，并且顾客不接受在上述两个时间段以外的时间 $[-\infty, a)$ 或 (b, ∞) 收货。

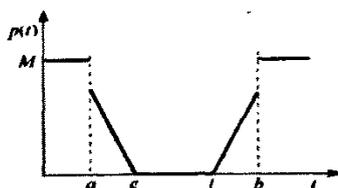


图 2-5 混合时间窗

此外，如果时间窗按形式来分有单边和双边时间窗之分，所谓单边时间窗是从双边时间窗中去掉了对最早服务时间的限制，客户不再要求车辆最早服务时间，而只要求在某个时间点之前为客户提供服务。

目前研究最多的是带双边时间窗的车辆路径问题，即只有在客户指定的时间窗口内为其提供服务。若车辆在客户指定的最早时间之前到达客户所在地，则需要等待直至到了客户要求的最早时间，才能对客户提供服务；此外，车辆必须在最晚服务时间之前为客户提供服务，否则不可行。

本文研究的 VRPTW 问题就是此类带双边时间窗的车辆路径问题。

2.5 VRPTW 问题的数学模型

本小节结合实际提出了本文的研究问题——带时间窗的车辆路径问题 (VRPTW)。建立了以车辆容量和客户需求等为约束条件, 以配送运输成本为目标函数的 VRPTW 数学模型。(2)

2.5.1 VRPTW 模型的建立

为研究问题的方便, 建立数学模型如下:

$$\min \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^K c_{ij} x_{ijk} \quad (2-14)$$

$$\text{s.t.} \quad \sum_{k=1}^K \sum_{j=1}^n x_{ijk} \leq K, i=0 \quad (2-15)$$

$$\sum_{j=1}^n x_{ijk} - \sum_{j=1}^n x_{jik} \leq 1, i=0, k \in \{1, 2, \dots, K\} \quad (2-16)$$

$$\sum_{k=1}^K \sum_{j=0}^n x_{ijk} = 1, i \in \{1, 2, \dots, n\} \quad (2-17)$$

$$\sum_{k=1}^K \sum_{i=0}^n x_{ijk} = 1, j \in \{1, 2, \dots, n\} \quad (2-18)$$

$$\sum_{i=0}^n d_i \sum_{j=0}^n x_{ijk} \leq Q, k \in \{1, 2, \dots, K\} \quad (2-19)$$

$$\sum_{i=0}^n \sum_{j=0}^n x_{ijk} (t_{ij} + s_i + w_i) \leq l_k, k \in \{1, 2, \dots, K\} \quad (2-20)$$

$$a_0 - s_0 - w_0 = 0 \quad (2-21)$$

$$\sum_{k=1}^K \sum_{i=0}^n x_{ijk} (b_j + s_i + t_{ij} + w_i) = b_j, j \in \{1, 2, \dots, n\} \quad (2-22)$$

$$e_i \leq (a_i + w_i) \leq l_i, i \in \{0, 1, \dots, n\} \quad (2-23)$$

其中 $x_{ik} = \begin{cases} 1, \text{车辆} k \text{ 经过边 } (i, j) \\ 0, \text{否则} \end{cases} \forall i, j \in N_1, \forall k \in V, i \neq j, i \neq n+1, i \neq 0$ 相关符号

除 2.2 节介绍的以外, 另记 $[e_i, l_i]$ 为点 $i, (i \in N)$ 的时间窗, 车辆必须在该时间窗内实施该点的配送服务。如果车辆在 e_i 之前到达点 i , 则必须等到 e_i 才能实施配送。为方便起见, 记点 0 为一条路线的起点, 点 $n+1$ 为一条路线的终点, 它们都表示同一中心库。并定义 $N_1 = N \cup \{0, n+1\}$, 如果一辆车的路线只有一条边 $(0, n+1)$, 则表示该车没有被启用。用 $-c_{0,n+1}$ 表示减少一辆车所节约的成本。

此外, 规定车辆的时间窗约束 $[e_0, l_0]$ 和 $[e_{n+1}, l_{n+1}]$, 其中 $[e_0, l_0]$ 表示车辆的发车时间窗, $[e_{n+1}, l_{n+1}]$ 表示车辆的返回时间窗。为了方便建模, 不失一般性地假设 VRPTW 的运输网络是完全连通的, 即每个点对都可通行。这种假设是合理的, 因为如果存在不可通行的点对, 记为 $(i, j) (i \neq j)$, 则可将成本 c_{ij} 取为充分大以使 VRPTW 的解不会选取该边。

式子 (2-14) 表示目标函数; 约束 (2-15) 表示最多有 K 辆车从中心库出发; 式子 (2-16) 表示每条路线开始结束于中心库; 约束 (2-17) 和 (2-18) 表示每个配送需求点恰好被一辆车访问一次; 约束 (2-19) 指每条路线的总需求量不得超过车辆装载容量; 约束 (2-20) 表示车辆的行驶时间限制, 这里指车辆时间窗口的宽度; 约束 (2-21) —— (2.23) 表示客户节点的时间窗约束。

2.5.2 VRPTW 模型的讨论

如果^[1]对所有需求点 i 有 $e_i = 0, l_i = +\infty$, 则 VRPTW 将退化为一个 VRP 问题。进一步, 如果 $|V| = 1$, 则退化为 TSP 问题。

如果取消约束 (2-19), 则 VRPTW 退化为一个 m-TSPTW 问题, 即带时间窗的多旅行商问题。进一步, 如果 $|V| = 1$, 则退化为 TSPTW 问题, 即带时间窗

旅行商问题。

如果取消约束 (2-17) 和 (2-18) (称为分配约束), 则 VRPTW 退化为一个关于每辆车的带时间窗口和容量约束的最短路问题 (Elementary Shortest Path Problem with Time Windows and Capacity Constraints, ESPPTWCC)。显然, 如果所有车辆的装载容量相同, 则每辆车的 ESPPTWCC 解都相同。

在模型中, 可以通过设置 $|V|$ 的大小来规定可使用车辆数的上限。如果 $|V|$ 足够大, 并且有 $c_{0,n+1} = 0$, 则表明车辆的使用数不受限制。也可以确保在解中不出现“空路线”。

如果取 $c_{0,n+1} = -M$ (M 为一个充分大的数), 则表示将车辆数作为 VRPTW 的第一目标。

为了支持 VRPTW 的优化算法, 需要假设成本与时间均满足三角不等式, 即 $c_{ij} \leq c_{ik} + c_{kj}$ 和 $t_{ij} \leq t_{ik} + t_{kj}$ 。三角不等式对于时间一般总是成立的, 但对于成本则未必成立。不过, 如果将所有 c_{ij} 都加上同一个数 C , 则只要 C 足够大, 总能满足三角不等式: $c_{ij} + C \leq c_{ik} + c_{kj} + 2C$, 而这样并不改变目标函数 (事实上, 只相当于在目标函数中增加一个常量)。因此, 三角不等式的假设不失一般性。

本章描述和定义一般车辆路径问题, 并针对实现中的约束条件, 对车辆路径问题的原型和各种衍生问题进行了简明的综述。重点对带时间窗的车辆路径问题进行了理论上的描述, 并建立了 VRPTW 的数学模型, 为后文的研究奠定了基础。

第 3 章 VRPTW 的求解方法

3.1 VRPTW 问题的时间复杂度、算法复杂性

对于优化问题的研究，其内容之一是其算法复杂性的研究。优化问题的算法复杂性可以通过转化为一个只要求回答是或否的判断问题来研究。用 P 表示能够在多项式时间内可以判断的问题集， NP 表示可以在非确定的多项式时间内可以判断的问题集(即可以用非确定图灵机在多项式时间作出判断)。

如果一个优化问题的算法在最坏情形下的运行时间为 $O(n^k)$ (其中 n 表示问题的规模大小，而 k 是一个不依赖于 n 的常数)，则称该算法为多项式算法。显然，该问题属于 P 类问题。

如果算法的时间复杂性函数不能用一个多项式界定，则称该算法为指数时间的算法。

对于一个问题 X ，如果任何一个 NP 类问题都可以在多项式时间归结为 X ，则当 X 被证明是 NP 类问题时，称 X 为 NP -完全问题，否则称 X 为 NP -难问题。这两类问题目前都没有多项式优化算法。

NP 完全问题是 NP 类中最难的问题，具有以下性质：任何一个 NP 完全都不能用任何已知的多项式算法求解；若任何一个 NP 完全有多项式算法，则一切 NP 完全问题都有多项式算法。如 0-1 背包问题、最大截问题、图着色问题都是典型的 NP 完全问题。现已证明的 NP 完全问题有上千个，但是没有找到任一问题的多项式算法。因此许多人认为 NP 完全问题不存在有效的多项式时间算法，但是尚未能给予证明。

对于 VRPTW 问题，已经被证明是一个 NP -难问题。事实上，求一个 VRPTW 可行解也是 NP -难问题，甚至比它有更少约束的 ESPPTWCC 问题也属于 NP -难问题。

对于 NP -难问题，目前使用数学规划的优化算法都是指数算法。特别的，对于一个 VRPTW 问题，有 $n^2 \times |V|$ 个决策变量，如果进行遍历搜索，则需要考察 $2^{n^2 \times |V|}$ 个不同的组合是否满足 $n^2 |V| + n |V| + 3 |V| + n + 1$ 个约束!这个计算量是相当大的，甚至对不超过 15 个点的规模也难以承受。

作为一个好的算法，应当在运算时间、求解质量、易实现程度、稳健性和可扩展性等 5 个方面具有较好的性质。所谓稳健性，是指算法不应对问题的一些特征差异表现得敏感，即一个稳健的算法应当对问题的各个实例都表现得相当稳定。而可扩展性，则是指一个算法应当能够灵活地适应模型中约束和目标函数的一些变化。

3.2 VRPTW 问题的算法研究

VRPTW 问题是 VRP 问题的推广。有关 VRP 问题的研究，在 1983 年 Bodin 就统计了约 700 篇研究文献^[7]。可是，带时间约束的车辆路线问题只在 20 世纪 80 年代才开始受到重视。作者认为这与信息技术的进步有着直接的联系。因为，在信息技术不发达的背景下，企业运营的节奏不快，时间约束不是重要的决策因素。

最早对 VRPTW 问题的算法研究是 1986 年 Solomon^[9]的启发式算法研究，1987 年 Kolen 等^[10]最早提出最优化算法：动态规划算法，这是第一个 VRPTW 最优化算法，该算法通过搜索树实施分支，应用动态规划求解带时间窗约束的最短路问题（SSPTW）确定下界。

1987 年 Solomon 较系统地将 VRP 问题的启发式方法推广应用到 VRPTW 问题^[11]，通过最坏情形的分析指出求解 VRPTW 问题从根本上比求解 VRP 问题更难。

1992 年，Desrochers 等结合线性规划松弛和列生成法提出一种 VRPTW 问题的最优化算法^[12]，成功求解了一些 100 个点规模的问题。

1994 年，Fisher 应用最小化 K-树方法提出 VRP 问题的一种最优化算法，并推广到求解 VRPTW 问题^[13]，并指出这类算法适用于求解时间窗口比较宽松的问题，并通过计算试验验证了这一点。

1997 年 Kohl^[14]通过 Lagrangian 松弛求出 VRPTW 的下界后，应用分支界定法对 VRPTW 进行优化求解，并成功求解了规模为 100 各点的问题。

随着信息技术的进步和优化算法的发展，除了上述的一些用于求解 VRPTW 问题的最优化算法外，更多用于求解 VRPTW 问题的是启发式算法，大致上可以分为两类：经典启发式算法（classical heuristics）和现代启发式算法（meta-heuristics）。大部分经典启发式算法是在 1960-1990 年间提出的，而现代

启发式算法是在 1990 年以后出现和发展起来的。

3.2.1 经典启发式算法(classic heuristics)

求解 VRP 的经典启发式算法大致上可分为三种类型: 路线构造启发式算法、路线改进搜索启发式算法、两阶段法。

路线构造启发式算法是在密切注意其费用的同时, 逐步构造一个可行解, 但其本身不包含对解改进的步骤。路线改进搜索启发式算法则试图通过在线路中或线路间进行一系列的边或点的交换来改进可行解。在两阶段法中, 问题被分解为两个阶段: ①将客户点进行分组, 以便能分别构成可行的线路; ②进行实际的线路构造, 确定其客户的访问顺序。两个阶段间可能会有一些反馈循环。两阶段法又可分为两种情形: 先分组后构造线路, 先构造线路后分组。在第一种情形中, 首先将客户点进行分组, 然后对每一组客户点, 构造一条车辆行驶路线。在第二种情形中, 首先构造一条包含所有客户点的环游, 然后将其分割为多条可行的车辆行驶路线, 最后试图通过在线路中或线路间进行一系列的边或点的交换来改进可行解。

本节首先综述 VRPTW 问题的路线构造算法, 再对路线改进的邻域搜索中采用的基本操作进行总结。

1、路线构造启发式算法

通常, 路线构造有两种实现方法, 即序列方法和平行方法。序列方法是指每次构造一条路线, 而平行算法则是同时构造多条路线。

1) 长回路算法 (Giant-Tour)

1986 年, Solomon^[9]应用“先路线后分组”(rout-first cluster-second)的思想提出一种长回路启发式算法。该算法首先生成一条长回路, 然后将它分解成许多较短的可行回路。在构造长回路时不需要考虑车辆容量与时间窗约束, 因此可应用 TSP 问题的算法。

2) Clarke-Wright 节约算法 (Clarke and Wright Savings Algorithm)

Clarke-Wright 节约算法最早由 Clarke 等^[15]提出并用于求解 VRP 的最著名的启发式算法。该算法是一种基于节约准则的逐步构解算法。其基本思想是: 将两条线路合并成一条时, 所带来的路程节约值越大说明合并越有效 (使总费用减少越多)。例如边 (i, j) 上的节约值定义为: $s_{ij} = c_{0i} + c_{0j} - \mu c_{ij}, \mu \geq 0$, 其中 c_{ij}

表示边 (i, j) 上的成本。当 $\mu = 1$ 时, 则 s_{ij} 表示将单独对点 i 和点 j 分别配送的两条路线改成由一辆车来完成点 i 和点 j 配送后所降低的成本。

节约算法是将每条路线只含一个配送点的 n 条路线作为初始解, 其中, 每条路线第一个和最后一个配送点分别称为路线的起点和终点。考察一条路线的起点与另一条路线的终点相连合并成新的一条路线。如果合并的路线满足约束条件, 则说这样的合并是可行的, 并将合并的节约值定义为连接这两条路线的边的节约值。选择节约值最大的可行合并进行一次路线的合并。当不存在可行合并时, 算法结束。

Solomon^[11] 建议在选择路线合并时对等待时间进行适当限制。设合并两条路线的连接边为 (i, j) , w_j' 表示在点 j 的等待时间, 则当 $w_j' > W$ (W 为参数) 时, 不选择该路线合并。

3) 序列插入法

由 Mole 和 Jameson^[20] 所提出的序列插入法 (Sequential Insertion Heuristics) 引入两个参数 λ 和 μ 来构造一条路线, 即

$$\alpha(i, k, j) = c_{ik} + c_{kj} - \lambda c_{ij}, \quad \beta(i, k, j) = \mu c_{ok} - \alpha(i, k, j)$$

算法步骤可以描述如下:

步骤 1 (初始化): 若所有顶点属于一条路线, 停止。否则, 构造一条原始路线 $(0, k, 0)$, 其中 k 是未确定路线的任何顶点。

步骤 2 (下一顶点): 对每个未确定路线的顶点 k , 以及原始路线的所有相邻点 r 和 s , 计算可行的插入费用 $\alpha'(i_k, k, j_k) = \min\{\alpha(r, k, s)\}$, 其中 i_k 和 j_k 是得出 α' 的两个顶点。若无可行的插入, 转步骤 1。否则, 插入原始路线的最好顶点 k' 由式子 $\beta'(i_k, k', j_k) = \max\{\beta(i_k, k, j_k)\}$ 来确定。

步骤 3 (线路优化): 用 3-opt 方法对当前线路进行优化, 转到步骤 2

其中, 两个参数 λ 和 μ 决定了几个标准的插入准则。例如, 若 $\lambda = 1$ 和 $\mu = 0$, 算法将把路程长度增加最少的顶点插入; 若 $\lambda = \mu = 0$, 被插入的顶点将对应于与两相邻顶点间距离之和最小者, 若 $\mu = \infty$ 和 $\lambda > 0$, 被插入的顶点将是距车场最远者。

4) 插入算法(push forward insertion heuristics, PFIH)

PFIH 算法的流程与序列插入法相似, 也是从初始路线出发, 序列构造路线。并在不存在可行插入时新增一条初始路线。插入算法的关键是选择最合适的未

分配点在路线中进行最佳位置的插入。

记当前路线为 (i_0, i_1, \dots, i_m) ，其中 $i_0 = i_m = 0$ ，对未分配点 u ，Solomon^[11]从距离和时间两方面，提出确定最佳插入位置的计算公式

$$c_1(i(u), u, j(u)) = \min\{c_1(i_{p-1}, u, i_p)\}, p = 1, \dots, m$$

对于未分配点 u^* 的选择，Solomon 提出以下选择公式：

$$c_2(i(u^*), u^*, j(u^*)) = \min\{c_2(i(u), u, j(u))\}, u \text{ 为未配送点, 且插入是可行的。}$$

其中， $c_1(i, u, j)$ 、 $c_2(i, u, j)$ 有三种计算方法：

$$1) c_{11}(i, u, j) = d_u + d_{uj} - \mu d_{ij}, \mu \geq 0; c_{12}(i, u, j) = b_{j_u} - b_j$$

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j), \alpha_1 + \alpha_2 = 1, \alpha_1 \geq 0, \alpha_2 \geq 0,$$

$$c_2(i, u, j) = \lambda d_{0u} - c_1(i, u, j), \lambda \geq 0.$$

其中 $c_{11}(i, u, j)$ 表示将点 u 插入相邻两点 i 和 j 之间的距离成本； $c_{12}(i, u, j)$ 表示由于点 u 的插入造成在点 j 的开始服务时间的延迟， $b_j = \max\{e_j, b_i + s_i + t_{ij}\}$ 为当前路线中 j 点的开始服务时刻， b_{j_u} 为插入点 u 后在点 j 新的开始服务时刻。

2) $c_2(i, u, j) = \beta_1 R_d(u) + \beta_2 R_t(u), \beta_1 + \beta_2 = 1, \beta_1 \geq 0, \beta_2 \geq 0$ ，其中 $R_d(u)$ 、 $R_t(u)$ 分别为插入点 u 之后的路线总长与路线总时间。

$c_1(i, u, j)$ 意义同 1) 中所述。

$$3) c_{11}(i, u, j)、c_{12}(i, u, j) \text{ 同 1) 中所述, } c_{13}(i, u, j) = l_u - b_u; c_2(i, u, j) = c_1(i, u, j);$$

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j) + \alpha_3 c_{13}(i, u, j), \alpha_1 + \alpha_2 + \alpha_3 = 1, \alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_3 \geq 0.$$

其中 $c_{13}(i, u, j)$ 表示点 u 在 i 和 j 之间插入的时间余地。

2. 路线改进搜索启发式算法

对于 VRPTW 解的改进，通常采用在解的某个邻域进行局部搜索的方法，而作为 VRPTW 问题的局部搜索算法，首先要对解的邻域进行定义，并制定路线

更新和算法终止的准则。一般地，一个解的邻域是指既定变换规则下将该解更新后所构成的解集。如果在解的邻域存在一个比该解更优的解，则进行解的更新，从而完成一次改进；否则，称当前解为局部最优解。这里，由两种选择更新的策略，即首先更新策略（first-accept, FA）和优选更新策略（best-accept, BA）。前者是指一旦出现更好的解，则即刻进行更新，而优选更新策略则要求从中选取一个最好的解来更换原来的解。

1) $k-opt$

如果将一条路线中的 k 条边用另外 k 条边替换，则这样的替换称为 k -替换 ($k-exchange$) 所有可能的 k -替换所形成新的路线集合构成一个搜索邻域。如果在搜索邻域内不存在目标值下降的解，则当前解为局部最优解，称为 k -最优解 ($k-opt$)。应用 k -替换进行路线改进的算法称为 $k-exchange$ 算法或 $k-opt$ 算法。

$k-opt$ 算法最早是由 Lin^[21]在 1965 年提出的。1977 年, Russe11^[22]中首先在带时间窗的车辆路线问题(具体地说, 是带边约束的 MTSP 问题)中应用了 $k-opt$ 算法, 并指出, 在时间窗约束较宽松的情况下这种算法的效果较好。1986 年, Baker 和 Schaffer^[23]首先将 $2-opt$ 和 $3-opt$ 推广应用到 VRPTW 问题。

图 3-1 描述 $2-opt$ 搜索的边替换方法。其中, 分别用两条边 (i, j) 和 $(i+1, j+1)$ 替换原路线中的两条边 $(i, i+1)$ 和 $(j, j+1)$, 并改变在点 $i+1$ 和点 j 之间的路线方向。

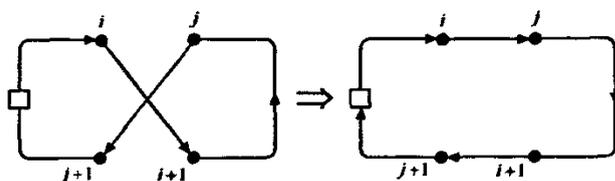


图 3-1 $2-opt$ 搜索的边替换方法

图 3-2 和图 3-3 描述了 $3-opt$ 搜索的边替换方法。在图 3-2 中, 分别用三条边 $(i, j+1)$ 、 $(k, i+1)$ 和 $(j, k+1)$ 替换原来路线中的三条边 $(i, i+1)$ 、 $(j, j+1)$ 和 $(k, k+1)$ 。在图 3-3 中分别用三条边 $(i, j+1)$ 、 $(k, i+1)$ 和 $(j, k+1)$ 替换原来路线中

的三条边 (i, j) 、 $(i+1, k)$ 和 $(j+1, k+1)$ ，并改变点 $i+1$ 和点 j 之间、点 $j+1$ 和点 k 之间的路线方向。

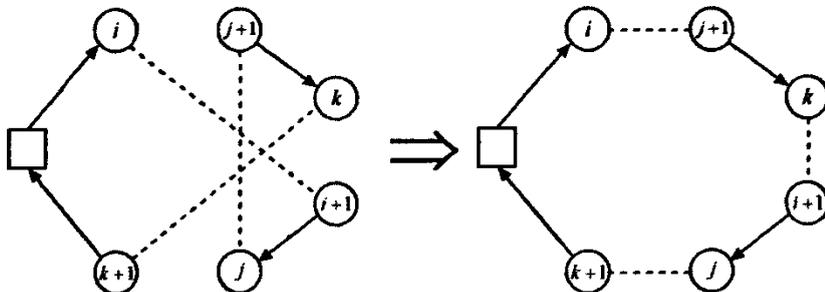


图 3-2 第一类 3-opt 搜索的边替换方法

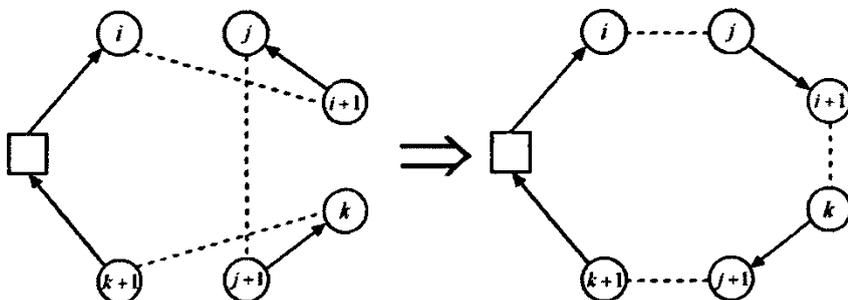


图 3-3 第二类 3-opt 搜索的边替换方法

2) 2-opt'

1995 年, Potvin 和 Rousseau^[24]提出一种两条路线之间的边替换方法—2-opt'法。其操作方法是,从两条路线中各取一条边,在不改变路线方向的前提下用新的两条边替换这两条边。图 3-4 描述了 2-opt'的边替换方法。图中用新的两条边 $(i, j+1)$ 和 $(j, i+1)$ 替换两条路线中的各一条边 $(i, i+1)$ 和 $(j, j+1)$ 。

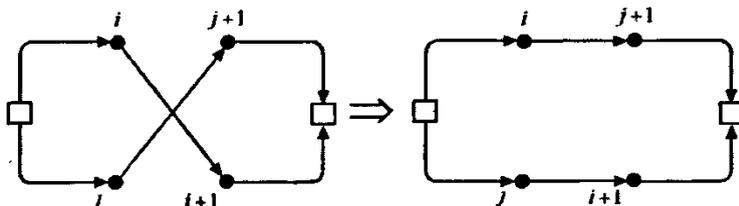


图 3.4 2-opt'的边替换方法

3) CROSS-exchange

1997年, Taillard 等^[25]提出了一种命名为 *CROSS-exchange* 的边替换方法。其操作是, 首先从一条路线中移除两条边 $(i-1, i)$ 和 $(k, k+1)$, 从另一条路线中也移除两条边 $(j-1, j)$ 和 $(l, l+1)$; 然后, 用新的 4 条边 $(i-1, j)$ 、 $(l, k+1)$ 、 $(j-1, i)$ 和 $(k, l+1)$ 替换。

4) Or-opt

1976年 Or 对 TSP 提出了一种以他的名字命名的 *Or-opt* 方法, 一个 *Or-opt-1* 替换是指改变一个点在一条路线的位置。而 *Or-opt* 方法是对 *Or-opt-1* 的推广, 即将一个点, 或者相邻的两个点或三个点作为一个整体改变其在一条路线中的位置。图 3-5 描述了 *Or-opt-2* 的替换方法, 点 i 和点 $i+1$ 是路线中相邻的两点, 将其作为一个整体移到点 j 之后, 并增加边 $(i-1, i+2)$ 后形成新的路线。从图中可以看出, *Or-opt* 是一种特殊的 *3-exchange* 操作。

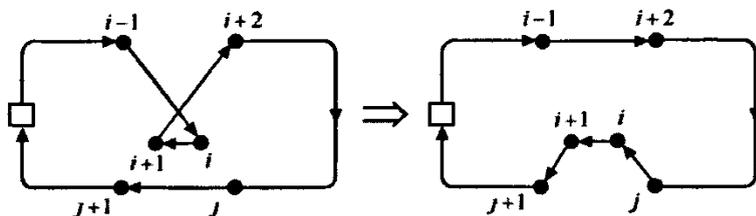


图 3-5 *Or-opt-2* 的边替换方法

1995年, Potvin 和 Rousseau^[24]将 *Or-opt* 方法推广应用到 VRPTW 的路线改进, 并结合使用其他边替换法。通过对测试问题的计算表明, 混合使用 *Or-opt* 和 *2-opt* 两种方法的计算效果要比 *3-opt* 算法好。

5) λ -interchange

1993年, Osman^[24]提出了一种 λ -interchange方法。设VRPTW的一个解为路线集 $S = \{r_1, \dots, r_p, \dots, r_q, \dots, r_k\}$, λ -interchange方法是指在一对路线 (r_p, r_q) 之间, 分别取点集 $S_1 \subseteq r_p$ 和 $S_2 \subseteq r_q$ (要求满足 $|S_1| \leq \lambda, |S_2| \leq \lambda$), 重新组合后构成新的两条路线 $r'_p = (r_p - S_1) \cup S_2$ 和 $r'_q = (r_q - S_2) \cup S_1$, 从而得到一个与 s 相邻的解 $S' = \{r_1, \dots, r'_p, \dots, r'_q, \dots, r_k\}$

6) Ejection Chains

1992年, Glover^[26]提出一种连锁排挤算法(Ejection Chains)。其操作是: 将一个点 c_j 从目标路线 r_n 中移出, 并插入另一条路线 r_m 。如果插入后的路线不满足约束, 则从路线 r_m 中移出一个点(即该点被排挤出来), 并将该点插入另一条路线。如此继续, 直到没有点需要被移出为止。

7) GENIUS法(GENERalized Insertion, UNstringing and STRINGing, GENIUS)

1992年, Gendreau等^[27]提出了一种GENIUS方法, 其基本操作是考察一个点的重新定位(relocation), 并在点的剥离(unstringing)和插入后立即对相应路线进行一次k-opt改进。其中的插入方法称为广义插入法(GENI)。图3-6描述了GENIUS方法的一次操作, 它分两个阶段完成。在第一阶段, 将点 i 从上部的路线中剥离出来, 添加边 $(i-1, i+1)$, 并对上部路线进行一次k-opt改进。在第二阶段, 将点 i 插入点 j 和 $j+1$ 之间, 接着对下部路线进行一次3-opt改进, 即用三条边 $(i, k), (k, j+1)$ 和 $(k-1, k+1)$ 替换 $(i, j+1), (k-1, k)$ 和 $(k, k+1)$ 。

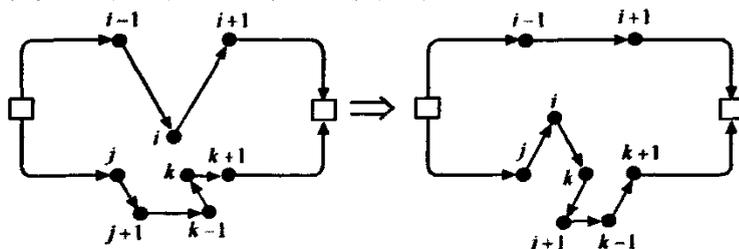


图 3-6 GENIUS 的一次操作

8) 循环k-转移法(cyclic k-transfers)

1993年, Thompson 和 Psaraftis^[28]提出一种循环 k -转移方法, 其中 k 是给定的整参数。对 VRPTW 的一个解 $S = \{r_1, \dots, r_j, \dots, r_k\}$, 记 $\{i_1, \dots, i_c\}$ 为 $\{1, \dots, k\}$ 的一个子集的排列 ($c \leq k$), 则循环 k -转移操作是将路线 r_j 中的 k 个点转移到路线 r_{j+c} 中, 其中, $j=1, 2, \dots, c$, $i_{c+1} = i_1$ 。特别地, 如果固定 c , 则称为 c -循环 k -转移方法。

可以看出对于路线的改进基本上是每次分别对一条车辆路线或几条路线中进行边替换操作, 有时候几种方法混合采用。

3. 两阶段法

本节我们首先介绍“先分组后构造线路法”, 再介绍“先构造线路后分组法”。其中, 前者又可分为三种类型: 初等分组法、舍项分枝定界法、花瓣算法。

1) 初等分组法

该方法对点集进行一次分组, 然后对每一组确定一条车辆行驶路线。下面介绍三种初等分组法(Elementary Clustering Methods): 扫描算法、基于广义指派算法和基于选址的启发式算法。后两种方法中假设车辆数为一个固定值。

(1) 扫描算法

扫描算法(Sweep Algorithm)是由 Gillett 和 Miller^[7]于 1974 年提出的。该算法可以有效地对付多达 250 点的问题。算法分两个阶段来构造一组解, 首先以车场为旋转中心转动一条射线将客户点按区域进行划分成组; 然后对每一区域内的客户点求解一个 TSP, 以确定一条车辆行驶路线。

1987 年, Solomon 将其推广应用于 VRPTW 问题的路线构造。所谓分组, 即分派给每辆车一组点。一种简单的分组方法是将以车站为原点的坐标平面划分为多个扇形区域, 并初步将每个扇形区域的点分派给一辆车。而所谓的“路线”, 是指在每个区域内, 采用扫描法选择未分配点, 然后应用插入算法扩充路线。如果在进行了一次“分组-路线”的路线构造后, 还存在未分配点, 则再进入“分组-路线”程序。如此反复, 直到所有点均已分配为止。

(2) Fisher-Jaikumer 算法

Fisher-Jaikumer^[31]算法也是非常著名的。与几何方法不同, 该算法通过求解一个广义指派问题(Generalized Assignment Problem, GAP)来分组。算法可描述如下:

步骤 1 (种子客户选择): 在 V 中选择种子客户点 j_k 以初始化每个组 k 。

步骤 2 (将其他客户分配到各种子客户): 计算把每个客户 i 分配给每个组 k 的费用, 即将客户 i 插入到车辆 k 从车场 0 直接到种子客户 j_k 并返回车场的线路的

$$\text{费用 } q_{ik} = \min\{c_{0i} + c_{ij_k} + c_{j_k 0}, c_{0j_k} + c_{j_k i} + c_{i0}\} - (c_{0j_k} + c_{j_k 0})$$

步骤 3 (广义指派): 以费用 q_{ij} , 客户权重 d_i , 以及车辆装载能力 C 求解一个广义指派问题。

步骤 4 (求解 TSP): 对与广义指派问题的解所对应的每一个区域(组), 求解一个 TSP。

在 Fisher-Jaikumer 启发式算法中, 车辆线路数(种子客户数) K 是预先确定的。

$$K \text{ 可由下式估计得到: } k = \left\lceil \sum_{i=1}^n d_i / C \right\rceil \text{ (取整数)}$$

种子客户可以用下列规则来选取:

①选择 $d_i > 0.5C$ 的客户作为种子客户, 因为这些客户中的任何两个都不能在同一条线路上。

②选择最远离车场的客户为第一个种子客户, 距离由车场和已确定的种子客户构成的集合最远的客户作为下一个种子客户, 然后按此规则继续进行, 直到选出的种子客户数达到所需要的车辆数。

(3) 基于选址的启发式算法

Bramel 和 Simchi-Levi^[32]描述了一种两阶段的、基于选址的启发式算法 (Location-based Heuristic)。算法中, 通过求解一个带能力限制的选址问题来确定种子客户, 而其余的客户则在第二阶段中逐步包括到所指派的线路中去。首先在 n 个客户点中选定 K 个种子客户, 并在确保指派给任一种子客户的总需求量不超过 C 的情况下, 使得所有客户到他们最近的种子客户的总距离为最小。然后, 在每一步, 将指派给该线路种子客户的、具有最小插入费用的客户插入线路中, 以构造该车辆线路。考虑一条由向量 $(0 = i_0, i_1, \dots, i_l, i_{l+1} = 0)$ 描述的不完全线路 k , 令 $T_k = \{0, i_1, \dots, i_l\}$, 记关于 T_k 的一个最优的 TSP 解的长度为 $t(T_k)$ 。则将一个线路外的客户 i 插入到线路 k 的插入费用 $q_{ik} = t(T_k \cup \{i\}) - t(T_k)$ 。由于精确地计算 q_{ik} 可能很耗费时间, 提出了两个近似值: 直接费用 $\bar{q}_{ik} = \min_{h=1, \dots, l} \{2c_{i_h}\}$ 以及最近插入费用 $\bar{q}_{ik} = \min_{h=0, \dots, l} \{c_{i_h} + c_{i_{h+1}} - c_{i_h i_{h+1}}\}$ 。由第一种规则定义算法是渐近最

优的。

2) 舍项分枝定界法

Christofides 等^[33]提出了一种求解带变量 K 的问题的舍项分枝定界法 (Truncated Branch-and-Bound)。该方法基本上是 Christofides^[34]的精确算法的一种简化。在这个算法中, 搜索树包含的层次与车辆线路数一样多, 在每一层都包含有一组可行但又不占绝对优势的线路。在所提出的如下执行步骤中, 搜索树是如此的简单以至于在每一层只由一个分枝组成, 因为在线路选择步骤中, 除了一个分枝外, 其它所有分枝都被剪枝。通过在每一层保留少数几条有价值的线路, 可以构造出一棵有限树。

从解的质量来看, 两阶段算法似乎要优于路线构造启发式算法。此外, 舍项分枝定界法一般能在较短的时间内求出比扫描算法更好的解。在多数情况下, Fisher-Jaikumer 算法的效果还是令人满意的。Bramel 和 Simchi-Levi 的基于选址算法通常能求出比 Fisher-Jaikumer 算法更好的解。

3.花瓣算法

扫描算法的一种自然扩展是生成几条线路, 称为花瓣(petals), 并通过求解如下形式的集划分问题来作出最后选择:

$$\begin{aligned} & \min \sum_{k \in S} d_k x_k \\ \text{满足} \quad & \sum_{k \in S} a_{ik} x_k = 1 \quad \forall i = 1, \dots, n, \\ & x_k \in \{0, 1\} \quad \forall k \in S, \end{aligned}$$

其中 S 是线路集合; $x_k = 1$ 当且仅当线路 k 在解中时; a_{ik} 为二进制参数, 当客户 i 属于线路 k 时, 其值等于 1; 而 d_k 为花瓣 k 的费用。如果线路对应于邻接的顶点区域, 则该问题可以在多项式时间内求解(Ryan 等^[35])。

该模型是由 Balinski 和 Quandt^[36]首先提出来的, 但是当 $|S|$ 较大时它就不实用了。Foster 和 Ryan^[37]、以及 Ryan 等^[35]提出了称之为 1-花瓣的启发式规则, 用于生成初始车辆线路的一个有价值的子集 S' 。Renaud 等^[38]通过不仅将单条车辆路线, 而且还有称之为 2-花瓣的、由两条嵌入的或相交的线路组成的构形包括在 S' 中, 从而把算法向前推进了一步。

Renaud 等^[38]将他们的 2-花瓣算法的效果与扫描法和 Foster 和 Ryan 等^[37]的 1-花瓣算法进行比较。对 14 个标准测试问题的运算结果表明, 平均来看, 2-花

瓣算法求出的解只高出已知的最好解的 2.38%(而扫描法是 7.09%) } 1-花瓣算法是 5.85%)。

4.先构造线路后分组法

在先构造线路后分组法(Route-First, Cluster-Second Methods)的第一阶段,是暂不考虑问题的约束条件,构造一个大的 TSP 环游;而在第二阶段,则将这个环游分解为可行的车辆线路。该算法适用于所需车辆数不确定的问题。该方法是由 Beasley^[39]首先提出的,他注意到第二阶段的问题是在非环形图上的一个标准的最短路径问题,这样就可以用 Dijkstra 算法在 $O(n^2)$ 时间内求解。在最短路径算法中,点 i 和 j 间的行驶费用 $d_{ij} = c_{oi} + c_{oj} + l_{ij}$, 其中 l_{ij} 是在 TSP 环游上从 i 到 j 的行驶费用。Haimovich 和 Rinnooy Kan^[41]指出如果所有客户具有单位需求量,该算法是渐近最优的。然而对于一般性的需求就不同了。目前还没有测试结果表明先构造线路后分组法优于其它方法。

3.2.2 现代启发式算法(meta-heuristic)

目前,在实际中使用的大多数标准的解构造方法和解改进方法都属于第一种类型(classical heuristics)。这些方法在解空间中进行非常有限的搜索,一般能在较短的时间内求出满意解。此外,这类算法中的大多数可以很容易地进行扩展,以对付在实际应用中所碰到的各种各样的约束条件。因此这类算法在商业软件包中仍然被广泛使用。

但相对于经典启发式算法,现代启发式算法不要求在每次迭代中均沿目标值下降方向,而允许在算法中适当接受目标值有所上升甚至不可行的解,其目的是能够跳出局部搜索邻域。它强调的是对解空间进行全局搜索,特别是对富有希望的区域进行纵深试探。这类方法一般将高级的邻域搜索技术、记忆结构、以及解的重组结合起来,所求出的解的质量比经典启发式算法的要高得多,但所付出的代价是较长的运算时间。此外,这类方法通常属于具体的问题类型,并需要一些经精心调试过的参数,这就有可能给将其扩展到其它情形带来困难。在某种意义上,现代启发式算法不过是高级的解改进方法,可以简单地被看作经典启发式算法功能的自然增强。然而,现代启发式算法使用了几个在经典启发式算法中未曾使用的新概念,而且常常将两者联合起来使用,实验证明这种结合可以取得较好解。

下面,对应用于 VRPTW 的现代启发式算法进行综述。

1、禁忌搜索算法 (tabu search, TS)

禁忌搜索算法最早由 Glover 在 1986 年提出,是局部搜索算法的扩展。该算法通过利用一个禁忌表记录已经到达过的局部最优点,并在后面的搜索中,根据某种限制循环的规则和禁忌表中记录的信息在当前搜索邻域中取一个合适的解。典型的禁忌是禁止在最近的 t 次迭代中选取某些边,以控制在最近的 t 次迭代中不会出现同样的解。有时,根据某种事先设定的规则,选用某个禁忌对象可能会使搜索邻域更快地接近全局最优点,这种规则被称为特赦规则(Aspiration Criteria)。

1994 年, Garcia 等^[16]首先将禁忌算法应用于 VRPTW 问题。其中,使用插入算法构造路线,在路线改进中运用 2-opt* 和 Or-opt 等算法操作。此后,文献中提出了各种禁忌搜索方法。

在禁忌搜索中,通过一些变化(diversification)与强化(intensification)规则来引导搜索将显著提高其搜索效率。1995 年, Rochat 和 Taillard^[18]定义了所谓的“适应性记忆信息”(adaptive memory),即在搜索过程中,将最好的解所包含的一些路线保存下来所汇集而成的路线信息。其目的是通过对适应性记忆信息中所有路线进行筛选与组合后,为禁忌搜索提供一个新的起始解。之后,在 1997 年, Taillard^[25]应用这种思想求解带软时间窗的 VRP 问题。

2.遗传算法(GENETIC ALGORITHMS Genetic algorithms GA)

遗传算法是借用适者生存规律进行局部搜索改进的一类算法。最优化问题的求解过程是从众多解中选出最优解,而生物进化的适者生存规律使得最具有生存能力的染色体以最大可能生存。二者的共同点使得遗传算法可以在优化问题中得以应用。具体体现在:1)进化发生在解的编码上,这些编码按照生物学术语称为染色体。由于对解进行了编码,优化问题的一切性质都可通过编码来研究;2)根据优化问题的目标函数定义适应性函数,按照适者生存的进化规律选取适应性大的个体交配;3)基因重组在保持遗传特性的前提下,允许适当的变异。

遗传算法最早是由 Holland 在 1975 年提出,并首先被 De Jong 用来解决复杂问题。该算法通过染色体的配对和变异过程实现种群的进化,每一次进化则对应解的一次迭代。当迭代次数达到最大次数限制或群体中的个体无显著差异时,迭代终止。

1991年, Thangiah^[18]首先将遗传算法用于求解 VRPTW 问题。该文将遗传算法用于点的分组。其中, 每一种分组方案被编码为一个染色体, 适应性函数值定义为根据相应的分组方案所构造的路线总成本, 而交配则指通过随机选取染色体的两个位串 (bit string, 即两个组) 进行的点的互换, 最后以小概率事件改变其中某些位的值来实现变异。

1999年, Homberger 和 Gehring^[29]提出了应用遗传算法求解 VRPTW 问题的进化策略。具体地, 将所谓的策略参数(strategy parameters)与解向量一起编码, 并通过重组和变异来实现进化。对于 VRPTW 问题, 策略参数包括某种局部搜索方法的使用频率, 以及在车辆数和总路长这两个目标之间交替改进的一个二进制参数。同年, Gehring 和 Homberger^[30]采用一种协同自治 (cooperative autonomy) 的方式在车辆数优化方法与总路长优化方法之间建立一种协作关系。具体地, 算法应用遗传算法最小化车辆数量, 应用禁忌搜索算法最小化总路长, 并通过制定解的更新规则建立这两个优化方法之间的协作关系。这是一种平行算法。2001年, Gehring 和 Homberger^[40]对这种方法提出了三点改进: 1) 在个体评价方面利用了车辆容量信息来确定是否删除某条路线; 2) 起用多个种群; 3) 达到车辆数下限的终止准则。

3、模拟退火算法(Simulated Annealing, SA)

1996年, Chiang 和 Russe11^[42]提出 VRPTW 问题的模拟退火算法。模拟退火算法实际上是一种随机松弛技巧, 它模拟了退火过程。在搜索的初始阶段, 算法跳向远点, 随着时间的延伸或“降温”, 跳跃幅度逐渐减小, 最终转向局部搜索下降方法。具体地, 设当前解为 S , 新解为 S' , 记 $\Delta = c(s') - c(s)$, 则选择 S' 替换 S 的条件是:

$$\Delta \leq 0 \text{ 或 } e^{-\Delta/T} > \text{random}(0,1)$$

其中, T 为参数, 称为“温度”, 其大小由事先制定的降温表控制, $\text{random}(0,1)$ 为在 0 到 1 之间的随机数。

2000年, Tan 等^[43]基于 2-interchange 法和单调下降的降温表提出一种快速模拟退火算法。当到达最低温度后, 通过参考初始温度和到达最好解时的温度设置一个新的温度, 然后重新启动模拟退火搜索过程。2001年, Li 等^[44]应用插入算法和扫除算法初始化路线后, 将邻域搜索方法与模拟退火程序相结合实现路线改进。

4、蚁群算法(Ant Colony Optimization, ACO)

蚁群算法最初由意大利学者 Dorigo M 于 1991 年提出,其本质上是一个复杂的智能系统,它具有较强的鲁棒性、优良的分布式计算机制,易于与其他方法结合等优点,在解决许多独自优化问题方面已经展现出其优异的性能和巨大的发展潜力,近几年吸引了国内外许多学者对其进行多方面的研究工作。

蚁群算法模拟了蚁群搜索食物的行为。在寻找食物时,蚂蚁会在它所经过的路径通过排放一种外激素(pheromone,在算法中称为信息素)作出标记,排放的量则根据路径长度和食物的等级决定。这些外激素为其它蚂蚁提供信息,并吸引他们前去搬运食物。

1999 年, Gambardella et al^[45]应用蚁群算法对 VRPTW 进行路径改进。首先构造两组相互协作的人工蚁群,第一个用来最小化车辆数,第二个用于最小化总路径长,并以共用解的方式建立协作关系。

5、导向性局部搜索方法(Guided Local Search, GLS)

1999 年, Kilby 等^[46]提出了一种导向性局部搜索法。其原理是:通过在成本函数中增加一个惩罚项,使搜索能够跳出局部搜索邻域。在惩罚项中,将对当前解与曾经搜索到的局部最优解之间存在的共性实施惩罚。具体地,作者对搜索过程中重复出现次数较多的边给予较大的惩罚。

6、GRASP 算法(Greedy Randomized Adaptive Search Procedure)

1995 年, Kontoravdis 和 Bards^[47]提出一种 GRASP 算法(Greedy Randomized Adaptive Search Procedure)。在路线构造过程中,根据 Solomon^[11]提出的成本函数,考察每个未分配点在每条路线中的最低成本的可行插入,计算每条路线的最低插入成本与总的最低插入成本之差,并对差求和,称为该点的惩罚值(penalty)。然后,按照惩罚值从大到小取 L 个未分配点,并从这 L 个点中随机选取一点扩充路线。每当完成了 5 次迭代后,便进行一次局部搜索改进。

7、综合启发式算法

2001 年, Braysy^[48]在一个算法中综合运用了多种搜索方法。该算法由四个阶段组成。在第一阶段,应用序列生成路线的方法构造一个初始解,并在每配送 k 个点后运用 Or-opt 法进行改进;第二阶段,应用连锁排挤法对车辆数予以

优化；第三阶段，应用 CROSS-exchanges 和插入方法对总路长予以优化；第四阶段，应用局部搜索方法，在不显著增加总路长和总车辆数的前提下，对等待时间予以优化。2002 年，Braysy 等^[49]作了进一步改进，如：在搜索过程中采用了阈值接受准则(Threshold Accepting)；将连锁排挤方法推广到多个点的转移(transfer)；在第三阶段，CROSS-exchanges 法作了修改，如变化路线的排列顺序，容许更多的插入位置；在第四阶段，混合使用了 CROSS-exchanges 和 GENIUS-exchanges 两种方法。

综上所述可以总结出以下几点：

采用精确算法虽然可以得到最优解，但耗时较多，且只是针对具体问题设计，不具备普适性；适用于解决有限规模的问题，不适用于现实中的大规模配送路径问题。

对路线的改进基本上是每次分别对一条车辆路线或几条路线进行操作，有时候几种方法混合采用。从解的质量来看，两阶段算法似乎要由于路线构造启发式算法。经典启发式算法中的大多数可以很容易地进行扩展，以对付在实际应用中所碰到的各种各样的约束条件。

作为经典启发式算法功能的自然增强的现代启发式算法，所求出的解的质量比经典启发式算法的要高得多。在某种意义上，现代启发式算法不过是高级的解改进算法。

本章总结求解 VRPTW 问题各种算法，包括最优化算法、启发式算法。其中启发式算法大致上可以分为两类：经典启发式算法(classical heuristics)和现代启发式算法(meta-heuristics)。其中启发式算法侧重于路线构造和路线优化，但是要得到一个好的解，通常要将几种算法混合起来使用，这也构成了本文算法的基础，下面的一章将重点介绍本文的算法及特点。

第 4 章 求解 VRPTW 问题的混合启发式算法

本章是本文的核心部分，提出了一种两步优化的实现策略。第一步，选取动态的 push forward insertion heuristics (stochastic PFIH) 算法产生机制来产生初始解；以大规模邻域搜索为新解产生机制，并对该方法进行了改进；最后，提出了一个混合模拟退火算法和大规模邻域搜索算法的混合算法。第二步，提出一种时间窗修正规则来调整时间窗，使得车辆在各个节点的等待时间为零，并给出了理论证明。

4.1 混合算法

本小节将给出产生初始解的方法和新解产生机制，以及混合算法流程。

4.1.1 初始解的产生机制

初始解的生成采用动态的 push forward insertion heuristics (PFIH) 算法。PFIH 是 Solomon^[11] 提出的一种计算插入新客户到现有路线中的有效方法，插入一个客户到路径中的可行性由以下方式检验：插入客户到现有路径的所有边，选择有最低的额外费用的边，当现有路线已满，就不再接受新的客户，PFIH 算法将产生新的路径，直到所有的客户被安排。通常第一条路径中的第一个客户的选取规则是距离中心库最近或是接受服务时间靠前者优先，但通过实验可以得知这种算法的缺点是：余下的客户将越来越难插入，产生了“creating skewed solution”现象，例如在开始阶段产生的路线有较多的客户，而在后面的阶段产生的路线中只有少量的客户。这样将导致车辆数较多，后面路线的车辆使用率不高，所以为了减少这种现象的产生，也为了提高初始解的质量，本文做出一些改进。首先，我们将车辆线路数 K' 预先确定的， K' 可由下式估计得到：

$$K' = \left\lceil \sum_{i=1}^n d_i / C \right\rceil \quad (\text{取整数}) \quad (4-1)$$

其中 d_i 为客户 i 的需求量， C 为车辆载重。如果初始解中的车辆数大于 K' ，则视该初始解无效，重新开始构造初始解。同时为了提高初始解的多样性，对于第

一个客户的选择采用随机的方式，这样就构成了动态的 PFIH 算法。具体的算法流程如下：

首先从所有客户中随机选择一个客户 i_1 ，作为第一条路径的第一个客户，开始构造第一条路线。在 (i_0, i_1) 中插入第二个客户，其中 $i_0 = 0$ ，从未分配的客户中选择一个节点 u ，由下面的公式(4-1)计算该点的最佳插入位置：

$$c_1(i(u), u, j(u)) = \min\{c_1(i_{p-1}, u, i_p)\}, p = 1, \dots, m$$

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j), \alpha_1 + \alpha_2 = 1, \alpha_1 \geq 0, \alpha_2 \geq 0, \quad (4-2)$$

其中， $c_{11}(i, u, j) = d_{iu} + d_{uj} - \mu d_{ij}, \mu \geq 0$ ，表示将点 u 插入相邻两点 i 和 j 之间的距离成本； $c_{12}(i, u, j) = b_{ju} - b_j$ 表示由于点 u 的插入造成在点 j 的开始服务时间的延迟， $b_j = \max\{e_j, b_i + s_i + t_{ij}\}$ 为当前路线中 j 点的开始服务时刻， b_{ju} 为插入点 u 后在点 j 的开始新的服务时刻。

对于未分配点 u^* 的选择，选择下面公式：

$$c_2(i(u^*), u^*, j(u^*)) = \min\{c_2(i(u), u, j(u))\},$$

$$c_2(i, u, j) = \lambda d_{iu} - c_1(i, u, j), \lambda \geq 0. \quad (4-3)$$

其中， u 为未配送点，且插入是可行的。

通过上述两个公式的计算，确定了插入第一条路径中的客户，重复这个过程，直到第一条路线中不存在可行的插入，此时新增一条路径，重复之前的过程，直到所有的客户都被安排，这样就得到了问题的一个可行解。

上述初始解的构造步骤可简述如下：

步骤 1：从所有客户中随机选择一个客户，构造一条初始路径。

步骤 2：按照公式(4.2) 计算未分配点的最佳插入位置。

步骤 3：按照公式 (4.3) 选择最佳未分配点，作为插入客户节点。

步骤 4：如果这条路径不再存在可行的插入，转向步骤 1；否则转向步骤 5。

步骤 5：输出初始解。

步骤 6: 判断初始解中的车辆数 K' 是否满足 $K' \leq \left\lceil \sum_{i=1}^n d_i / C \right\rceil$, 若满足, 输出初始解; 否则转向步骤 1.

4.1.2 新解产生机制

本文采用大规模邻域搜索方法产生新解, LNS 是 1998 年由 Pawl Shaw 提出的, 它从初始解出发, 经过 re-move (从当前解 A 中移走部分客户, 得到 B) 和 re-insertion (将移走的客户重新插回 B 中) 两个过程的不断优化, 最终得到最优解。

基于 VRPTW 问题的特殊性, 将 LNS 算法中的 re-move 过程和 re-insertion 过程具体描述如下:

在 re-move 过程中, 用符号 A 表示当前解, c 表示将要被移走的客户, R 表示被移走的客户组成的集合, p 表示移走的客户数目, B 表示从 A 中移走 p 个客户后剩余的部分解, 策略如下:

从 A 中随机移走一个客户 Z 作为 R 中的第一个元素; 取 A 中剩余的客户与 Z 的相关性最大的客户为 R 中的第二个元素; R 中其余的元素由 A 中剩下客户与此前集合 R 的相关性的大小来确定, 重复该过程 $p-2$ 次, 直到所需元素都选好。

其中, 两个元素 i, j 间的相关性以及元素 i 与集合 R 总体相关性定义如式 (4-4)、(4-5)

$$r(i, j) = 1 / (t_{ij}' + v_{ij}) \quad (4-4)$$

$$R(i, R) = \sum_{j \in R} r(i, j) \quad (4-5)$$

式中, $t_{ij}' = t_{ij} / \max t_{ij}$, $v_{ij} = 0$, 若 i, j 在同一条路径上, 否则为 1。

为了将 re-move 过程之后产生的部分解变成可行解, 要采用 re-insertion 过程, 将移走的客户集 R 中的元素重新插回部分解 B , 以产生更优解。但当 R 中的所有客户插回 B 中时会有很多种插入组合, 为使插入可行, 设计以下算法框架。

首先计算 R 中每个客户的最佳插入位置: 对客户而言, 在多个插入位置中, 选择使目标函数值增加最少的那个位置即为该客户的最佳插入位置, 并记下该位置对应的目标值。然后, 将每个客户最佳插入位置对应的目标值进行排

序, 选择目标值增加最少的客户 c , 将 c 插回部分解 B 中, 把 c 从 R 中移走, 同时搜索将 R 中剩余的客户都插回 B 能得到的所有解; 重复以上过程, 直到 R 为空集, 即当初被移走的客户都已经插回 B 中, 比较 B 与目前找到的最优解, 若 B 比目前最优解要好, 则 B 成为当前最优解。

4.1.3 VRPTW 混合算法流程

本文提出的混合算法, 在第一阶段充分利用了 PFIH 算法的特点, 为路线优化阶段提供高质量的初始解; 在第二阶段为了扩展解的搜索空间, 避免搜索过程中陷入局部最优, 将 SA 算法和 LNS 算法进行混合, 实现路线优化, 寻找最优解。混合算法流程具体如下:

步骤 1: 采用 PFIH 产生初始解 S_0 , 并计算目标函数值 $f(S_0)$ 。给定初始温度 T_0 , 每个温度的迭代次数为 l , 最大迭代次数为 L , 到目前为止连续不接受新解的次数为 m , 连续不接受新解的最大次数为 M 。

步骤 2: 若满足结束条件: $T_k > T'$ (T' 为终止时的温度), 则转入步骤 3; 否则转入步骤 6。

步骤 3: 若 $l < L$, 且 $m < M$, 则转入步骤 4; 若 $m > M$, 转入步骤 6。

步骤 4: 用 LNS 产生新解, 并计算 $\Delta f = f(S') - f(S_0)$ 。若 $\Delta f < 0$, 则接受新解作为新的当前解, 转入步骤 3。若 $\Delta f > 0$, 则以概率 $e^{-\Delta f/T}$ 接受新解为当前解, 若接受, 则 $m = 0$, 转入步骤 3; 否则, $m = m + 1$, 转入步骤 3。

步骤 5: T 逐渐减少, 以公式 $T_{k+1} = \alpha T_k, k \in N$, 转入步骤 2。

步骤 6: 输出最优解, 算法结束。

在约束条件非常复杂的问题中, LNS 具有竞争力强的搜索技巧, 但是它对初始解的要求比较高。为了更快搜索到最优解, 本文采用有效的 PFIH 算法来构造初始解, 并将具有较强全局优化能力的 SA 与 LNS 结合来优化初始解。同时提出一种调整车辆时间窗口的算法, 具体如下节:

4.2 回归迭代策略

由于可行路径不仅要使车辆行驶路径最小, 还应使车辆总等待时间尽可能小, 甚至为零。所以本文在将最小化车辆行驶距离作为首要优化目标的同时,

也考虑到车辆运输过程中的等待成本，通过调整各节点的时间窗，在每次得到可行解之后计算出每辆车的最佳出发时间，以使其等待时间为零。本文将此调整过程称为回归迭代策略，具体如下：

记第 K 辆车的路线为 $0 \rightarrow v_1 \rightarrow \dots \rightarrow v_n \rightarrow 0$ ，其中 v_i 的时间窗为 $[e_{v_i}, l_{v_i}]$ ，调整后为 $[e'_{v_i}, l'_{v_i}]$ ， $v_i \in \{1, 2, \dots, n\}$ ，整条路线中各节点的时间窗调整步骤为：

步骤 1: 记 $[e_0, l_0]$ 为车辆的时间窗。

步骤 2: 计算 $e'_{v_n} = \max\{e_{v_n}, e_0 - t_{0,v_n} - s_{v_n}\}$ $l'_{v_n} = \min\{l_{v_n}, l_0 - t_{0,v_n} - s_{v_n}\}$ 。

步骤 3: 更新 v_n 的时间窗为 $[e'_{v_n}, l'_{v_n}]$ ，重复步骤 2 计算 v_{n-1}, \dots, v_1 的新时间窗。

步骤 4: 记 $e_k = e'_{v_1} - t_{v_1,0}$ ， $l_k = l'_{v_1} - t_{v_1,0}$ ， $[e_k, l_k]$ 即为车辆 K 从中心库出发的时间窗。

由以上回归迭代策略的步骤可知，对任意的 e'_{v_i} ， l'_{v_i} 都有 $e'_{v_i} \leq l'_{v_i}$ 成立，不然，所得最优解是违背时间窗的不可行解。为了证明该策略的可行性，即经过调整客户时间窗口计算出的车辆出发时间，可使该车辆的等待时间为零。下面给出定理 4.2 及其证明。

定理 4.2: 对于同一路径中的任意两相邻客户，在新的时间窗口下，都可使车辆在这两个客户处的等待时间为零。

证明：假设客户 i, j 是两相邻客户（其中 i 在 j 之前被访问），都由车辆 K 提供服务。并假设调整后，各自的新时间窗为 $[e'_i, l'_i]$ 、 $[e'_j, l'_j]$ ，其中 b'_i 、 b'_j 分别为客户 i, j 新的开始服务时刻， w'_i 、 w'_j 分别为车在客户 i, j 处的新的等待时间，则有 $e'_i \leq b'_i \leq l'_i$ 、 $e'_i = \max\{e_i, e'_j - t_{ij} - s_i\}$ ， $l'_i = \min\{l_i, l'_j - t_{ij} - s_i\}$ 。

显然在新时间窗口下，服务客户 i, j 可行，下面用数学归纳法证 $w'_i \leq 0$ 、 $w'_j \leq 0$ 。

(1) 若 i 为车辆 K 服务的第一个客户，显然有 $w'_i = 0$ ，此时 j 为车辆 K 服务的第二个客户，而 $w'_j = \max\{0, e'_j - (b'_i + s_i + t_{ij})\}$ ，其中 $e'_j - (b'_i + s_i + t_{ij}) =$

$(e_j' - s_i - t_y) - b_i' \leq e_i' - b_i' \leq 0$, 即 $w_j' \leq 0$ 。

(2) 若 i 为第 m 个客户, j 为第 $m+1$ 个客户, 且有 $w_i' \leq 0$ 、 $w_j' \leq 0$ 成立。

则当 i 为第 $m+1$ 个客户, j 为第 $m+2$ 个客户时, 由上面推导过程 (1)、(2) 可知 $w_i' \leq 0$ 和 $w_j' \leq 0$ 都成立。

综上, 当 i 为车辆 K 服务的任意一个客户都有 $w_i' \leq 0$ 、 $w_j' \leq 0$ 成立, 定理得证。

第 5 章 算法实现及性能分析

5.1 试验数据的选取

为了验证本文算法的性能,本文选择 Solomon 的标准数据中的 C101^[12]系列数据进行实验验证。此类问题将点分为多个组,每组的点坐标由服从均匀分布的伪随机数产生,因为问题的决策周期较短,因而每条路线只有 5 到 10 个点。数据格式具体见附录 1, C101 中一共有 100 个客户(其中编号为 1 的客户为配送中心,2—101 为需要服务的客户),配送中心和客户的 x 与 y 坐标值都在 $[0,100]$ 之间。定义客户的最早服务时间都大于配送中心的最早上班时间,最晚服务时间小于配送中心的最晚下班时间,同时还要满足所有客户对货物的需求均不能超过车辆的容量限制。

为了编程分析该试验数据,需要进一步的具体化解决的问题,假设一个车场(配送中心)为 100 个位于不同地理位置、且对送货时间有不同要求的顾客提供服务。其中所有车辆的型号相同,载重量均为 200 单位;车辆在所有顾客处的服务时间均为 90 单位(时);车辆的速度为 1 个单位,本文将车辆的时间窗定义为 $[0, 1236]$,即车辆最早 0 时出发,1236 时必须回到配送中心。

5.2 试验结果

按照 4.1 中介绍的动态 PFIH 算法得到问题的初始解如表 5-1。

表 5-1 C101 问题的初始解

序号	客户数	客户的访问次序	路径长
1	12	0-20-24-25-27-29-30-28-26-23-22-21-47-0	78.19677
2	11	0-67-65-63-62-74-72-61-64-68-66-69-0	59.40312
3	12	0-5-3-7-8-10-11-9-6-4-2-1-75-0	59.62804
4	12	0-43-42-41-40-44-46-45-48-51-50-52-49-0	63.98909
5	10	0-90-87-86-83-82-84-85-88-89-91-0	76.06957
6	9	0-13-17-18-19-15-16-14-12-99-0	121.7607
7	8	0-98-96-95-94-92-93-97-100-0	95.48098

8	9	0-32-33-31-35-37-38-39-36-34-0	97.22717
9	8	0-57-55-54-53-56-58-60-59-0	101.8826
10	9	0-81-78-76-71-70-73-77-79-80-0	127.2975
合计	100		880.9355

按照 4.2 中的混合算法得出问题的最优解如表 5-2, 算法中各参数值分别为:
 $T_0=100$, $T'=0.8\alpha=0.8$, $l=5$, $L=30$, $m=20$, $M=30$ 。

表 5-2 C101 问题的最优解

序号	客户数	客户的访问次序	路径长
1	11	0-20-24-25-27-29-30-28-26-23-22-21-0	50.80355
2	11	0-67-65-63-62-74-72-61-64-68-66-69-0	59.40312
3	12	0-5-3-7-8-10-11-9-6-4-2-1-75-0	59.61804
4	13	0-43-42-41-40-44-46-45-48-51-50-52-49-47-0	64.80748
5	10	0-90-87-86-83-82-84-85-88-89-91-0	76.06957
6	8	0-13-17-18-19-15-16-14-12-0	95.88471
7	9	0-98-96-95-94-92-93-97-100-99-0	95.94313
8	9	0-32-33-31-35-37-38-39-36-34-0	97.22717
9	8	0-57-55-54-53-56-58-60-59-0	127.2975
10	9	0-81-78-76-71-70-73-77-79-80-0	101.8826
合计	100		828.9369

为了验证本文算法的有效性, 将它与其他算法的结果做了比较, 比较结果见表 5-3。

表 5-3 本文设计的算法和其他算法的比较 (C101)

Algorithm	Number of vehicle	Total distance	Reference
PFIH	10	853	Solomon M.M.(1983)
ACS	10	833	Xuan Tan et.al (2005)
GIDEON	10	833	Thangiah et.al (1993)
CGH	10	828.94	Alvarenga.G.B.(2004)
mGA	10	828.94	K.C.Tan et. al (2001)
Best Result	10	827.3	Desrochers et.al (1992)
Our hybrid algorithm	10	828.937	Jiangwx et.al (2007)

按照 4.2 节介绍的回归迭代策略算出的每辆车在各节点的最佳出发时间如下, 经过这样的处理, 可以使车辆的总等待时间均为零。

路径 1 中车辆在各节点的最佳出发时间:

21.3944 121.394 216.394 308.394 494 589 682 774
867 960 1052 1062.2

路径 2 中车辆在各节点的最佳出发时间:

20.7934 123 214 306 401 494 589 682 774 867 962.385
1054.39 1070.2

路径 3 中车辆在各节点的最佳出发时间:

21.1933 126.326 217.326 309.326 402.154 495.76 588.76 681.922
774.158 866.394 960 1052 1145 1160.81

路径 4 中车辆在各节点的最佳出发时间:

0.220282 106.773 199.773 291.773 383.773 476.773 569.602 661.602
753.602 846.602 938.838 1032 1125 1217 1235.03

路径 5 中车辆在各节点的最佳出发时间:

24.3845 135 230 321 417 510 605.831 698.659 791.659 884.488
978.093 1000.45

路径 6 中车辆在各节点的最佳出发时间:

20.5979 141.404 236.789 328.789 422.394 516 608 703 798 893
926.54

路径 7 中车辆在各节点的最佳出发时间:

17.9921 139.615 231.615 327 422 517.831 609.831 704.831 799
798 893 926.541

路径 8 中车辆在各节点的最佳出发时间:

24.1807 161.615 254.615 346.615 536.615 629.615 723.615 814.615
910 961.478

路径 9 中车辆在各节点的最佳出发时间:

21.1942 142 236 329 424 519 614 706 799 837.079

路径 10 中车辆在各节点的最佳出发时间:

24.6148 149.615 241.615 336.615 432 526 618 711 811.44 846.497

5.3 结果分析

对比表 5-1 和表 5-2, 可以看出本文第二阶段混合算法优化初始解的效果是明显的。当然, 表 5.2 所列出的实验结果是多次运行结果其中之一, 其他结果与本结果可能不同, 但是优化初始解的趋势都是明显的。产生不同结果的原因主要在于: 初始解的第一个客户的选择是随机的, 同时, 在采用大规模邻域搜索过程中, **remove** 过程中的客户的选择也是随机的, 这些运算过程都需要随机产生相应的随机数, 而随机数的好坏也在一定程度上影响着大规模邻域搜索的搜索性能, 所以对于同样的参数集合也可能会得到不同的解。此外由于不同参数解的性能往往不同, 因此一般需要反复调整或设定参数, 使得搜索到的解产生了很大的不确定性。

从表 5-3 中结果对比显示本文的混合算法相比于其他算法效果较明显, 与当前最优解的结果相差仅 0.197%, 表明本文算法有待进一步改进以提高解的质量; 此外算法的适用范围也有待研究。

对于 100 个客户且带有时间窗的大规模复杂运输问题, 如果仅凭决策者的经验, 是很难在较短时间内做出一个合理的运输路径规划的, 即使推断出一个结果也往往会产生很多负面的影响, 极大的提升了相关决策费用。另外带时间窗的车辆路径问题是 NP 完全问题, 本文采用基于模拟退火算法和大规模邻域搜索等启发式方法, 来解决大客户数目 (>100) 的车辆路径问题可以在相对短的时间内得到一个满意解。因此对车辆路径问题的研究并将其应用与决策支持中, 有利于降低实际运输问题中的成本, 提高工作效率, 具有一定的实际应用价值。同时随着计算机处理器能力的提高和搜索算法的进一步改进, 这一时间还将缩短。

现代物流配送业和供应链的发展, VRPTW 已成为各企业、物流公司等行业亟待解决的问题。因而研究高效的智能启发式算法对于提高物流效率、降低物流成本, 创造社会价值有着重大的意义。本文为求解该问题提出了一种结合模拟退火算法和大规模邻域搜索算法的混合式算法, 该算法包含了两者的优点, 同时也给出了一种回归迭代策略算出每辆车的最佳出发时间, 使等待成本为零。实验结果表明, 本文提出的算法不仅可以扩展解的搜索空间, 跳出局部解, 还具有收敛速度快的特点, 不失为一种有效的算法。

第 6 章 总结与展望

本文结合配送系统中对车辆路径问题的要求,提出了本文的研究内容——带时间窗的车辆路径问题(VRPTW)。首先详细描述了车辆路径问题,然后针对 VRPTW 问题进行数学建模,介绍了求解车辆路径问题的算法的研究内容、发展方向,并在此基础上提出了一种混合算法,并将此算法用于著名的 Solomon 标准测试数据中的 C101,通过对试验结果的分析 and 讨论,本文得到以下几点结论:

1、对 VRPTW 问题进行了综述。描述了 VRPTW 问题的概念,介绍了它的历史和研究现状、数学模型、时间复杂度。以综合性能的角度对求解 VRPTW 问题的算法进行了一定的归纳和分析。

2、采用动态的 PFIH 算法来构建问题的初始解,保证了初始解的多样性,同时在产生的初始解中设置了限制条件,实现对初始解的筛选,为第二阶段路径优化提供了高质量的初始解。

3、将模拟退火算法和大规模邻域搜索算法混合进行路径优化,得到问题的最优解,充分利用了两种算法的优点,克服了它们的缺点。

4、提出一种回归迭代策略调整车辆和客户的时间窗口,使得车辆在每个客户的等待时间为零,真正实现了路径的优化,节省了成本。

5、由前一章的结果分析可知,本文的混合算法求解效果比较好,得到了满意解。

在本文研究基础上,还可以进一步开拓以下几个方面的工作:

1) 带时间窗车辆路径问题的延伸,可以从以下几个方面进行考虑:

①可以将问题转化为带软时间窗的车辆路径问题,或是带回送的带时间窗的车辆路径问题,或是随机 VRPTW(包括随机顾客和随机需求量)等等,这样更贴近实际的运输配送问题。

②转化为开放式的带时间窗的车辆路径问题,更人性化,也更适合那些没有专门配送车队需要租车使用的物流公司。

③当信息条件不是很清晰的时候可以转化为模糊 VRPTW,其中包括具有模糊需求信息和模糊旅行时间,及模糊预约时间等。

④将配送中心理解为供应链中的一个环节时,又可转化为多中心库的

VRPTW 问题；也可以与配送中心的库存联合起来，形成库存-运输联合优化问题。

2) 在算法方面的改进，从算法实现的角度来讲，本文采用的混合算法虽然能在 Solomon 标准数据中的 C101 数据取得满意解，但仍存在一些缺陷。在求解过程中，得到最优解的频率不是很高，这与产生初始解的随机性强有关，同时，在采用大规模邻域搜索产生新解的过程中费时较多，有待改进，以节省运算时间；其次，算法在其他数据中的测验结果是否也让人满意，还有待研究，或者将算法应用于其他优化问题，以期找到该算法的最佳应用范围。

参考文献

- [1] 戴锡. 车辆路线问题的二阶段启发式算法及其在现代物流配送中的应用. 博士论文. 上海: 复旦大学, 2004
- [2] 张文修, 梁怡. 遗传算法的数学基础. 西安交通大学出版社. [M]2000.
- [3] 郭耀煌等. 运筹学与工程分析. 中国建筑工业出版社, [M]2002, 188~245.
- [4] Taillard, E., P.Badeau, M.Gendreau, F.Guertin, J. Y. Potvin. A Tabu Search Heuristics for the Vehicle Routing with Soft Time Windows. *Transportation Science*, 1997, 31(3):170~168.
- [5] DANTZIG G B, RAMSER K B. The truck dispatch Problem *Management Science*[J]. *Management Science*, 1959; 12(1): 80~91.
- [6] Christofides N, Mingozzi A, Toth P. The Vehicle Routing Problem[C]. *Combinational Optimization*, New York: Johnly Wiley, 1979.
- [7] L.Bodin, B.L.Golden, A.Assad and M.Bal. Routing And Scheduling of Vehicles And Crews: The State of The Art. *Computers & Operations Research*, Vol.10, No. 2: 63~193.
- [8] E.K.Baker(1983). An Exact Algorithm For The Time-constrained Traveling Salesman Problem. *Operations Research* 1983(31):938~945.
- [9] M.Solomon. On the Worst-Case Performance of Some Heuristics for the Vehicle Routing and Scheduling Problem with Time Window Constraints. [J] *Networks* 1986(16): 161~174.
- [10] AWJ Kolen, Kan AHG Rinnooy and HWJM Trienekens. Vehicle Routing With Time Windows. [J] *Operations Research* 1987(35):266~273.
- [11] M.M.Solomon. Algorithms For The Vehicle Routing And Scheduling Problems With Time Windows Constraints. [J] *Operations Research* 1987(35):254~265.
- [12] M.Desrochers, J.Desrosiers and M.M.Solomon. A New Optimization Algorithm For The Vehicle Routing Problem With Time Windows. [J] *Operations Research* 1992(40):342~354.
- [13] M.L.Fisher. Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees. [J] *Operations Research* 1994(42):626~642.
- [14] N.Kohl and OBG Madsen. An Optimization Algorithm For The Vehicle Routing Problem With Time Windows Based On Lagrangian Relaxation. [J] *Operations Research* 1997(45):395~406.
- [15] G. Clarke and J.V. Wright. Scheduling of vehicles from a central depot to a number of delivery points. [J] *Operations Research*, 1964(12): 568~581.
- [16] B.L.Garcia, J.Y.Potvin and J.M.Rousseau. A Parallel Implementation of The Tabu Search Heuristic For Vehicle Routing Problems With Time Windows Constraints. [J] *Computers & Operations Research*, 1994(21):1025~1033.

- [17] Y. Rochat and E. Taillard. Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. [J]Journal of Heuristics,1995(1): 147~167.
- [18] S.R.Thangiah, K.E. Nygard and P.L.Juell. GIDEON:A Genetic Algorithm System for Vehicle Routing with Time Windows. [J]Proceedings of Seventh IEEE Conference on Artificial Intelligence Applications, IEEE Computer Society Press, Los Alamitos.1991, 322~328.
- [19] R.H.Mole and S.R.Jameson. A sequential route-building algorithm employing a generalized savings criterion. [J]Operational Research Quarterly,1976 (27):503~511.
- [20] B.Gillett and L.R.Miller. A heuristic algorithm for the vehicle dispatch problem. [J]Operations Research, 1974(22):340~349.
- [21] S.Lin.Computer Solutions of the Traveling Salesman Problem. [J] Bell System Technical Journal ,1965(44):2245~2269.
- [22] R.Russell .An Effective Heuristic for the M-tour Traveling Salesman Problem with Some Side Conditions. [J] Operations Research,1977(25): 517~524.
- [23] E.K.Baker and J.R.Schaffer. Solution Improvement Heuristics for the Vehicle Routing and Scheduling Problem with Time Window Constraints. [J] American Journal of Mathematical Management Sciences , 1986(6):261~300.
- [24] J.Y.Potvin and J.M Rouseau.An Exchange Heuristic For Routing Problems With Time Windows. [J]Journal of Operational Research Society.1995(46): 1433~1446.
- [25] E.Taillard et al.A Tabu Search Heuristic For The Vehicle Routing Problem With Soft Time Windows. [J] Transportation Science, 1997(31):170~186.
- [26] F.Glover.New Ejection Chain and Alternating Path Methods for Traveling Salesman Problems. [J]Computer Science and Operations Research: New Developments in Their Interfaces,O.R. Balci Sharda, and S. Zenios (eds), Pergamon Press, Oxford .1992: 449~509.
- [27] M.Gendreau, A.Hertz and G.Laporte. New Insertion And Postoptimization Procedures For The Traveling Salesman Problem. [J] Operations Research, 1992(40):1086~1094.
- [28] P.M.Thompson and H.N. Psaraftis. Cyclic Transfer Algorithms for Multivehicle Routing and Scheduling Problems. [J]Operations Research,1993(41): 935~946.
- [29] J.Homberger and H.Gehring.Two Evolutionary Metaheuristics For The Vehicle Routing Problem With Time Windows. [J] INFOR1999(37):297~318.
- [30] H.Gehring and J. Homberger. A Parallel Hybrid Evolutionary Metaheuristic for the Vehicle Routing Problem with Time Windows. [J]Proceedings of EUROGEN99, K.Miettinen, M.Makela and J. Toivanen (eds), University of Jyvaskyla, Finland. 1999,57~64.
- [31] M.L. Fisher and R. Jaikumar. A generalized assignment heuristic for the vehicle routing problem".[J] Networks, 1981(11):109~124.
- [32] J. Bramel and D. Simchi-Levi. A location based heuristic for general routing problems. [J]Operations Research, 1995(43):649~660.

- [33] N. Christofides, A.Mingozzi, and P.Toth. The vehicle routing problem. In: N. Christofides, A.Mingozzi, P.Toth,et al. eds. *Combinatorial Optimization*, Chichester: Wiley, 1979. 315~338.
- [34] N.Christofides. The vehicle routing problem. [J]RAIRO, 1976(10):55~70.
- [35] D.M. Ryan, C. Hjorring, and F. Glover. Extensions of the petal method for vehicle routing. [J] *Journal of Operational Research Society*, 1993(44): 289~296.
- [36] M. Balinski and R.Quandt. On an integer program for a delivery problem. [J]*Operations Research*, 1964(12): 300~304.
- [37] B.A. Foster and D.M. Ryan. An integer programming approach to the vehicle scheduling problem. [J]*Operations Research*, 1976(27): 367~124.
- [38] J. Renaud, F.F. Boctor, and G. Laporte. An improved petal heuristic for the vehicle routing problem. [J] *Journal of Operational Research Society*, 1996(47):329~336.
- [39] J.E.Beasley. Route-first cluster-second methods for vehicle routing. [J] *Omega*, 1983(11):403~408.
- [40] H.Gehring and J.Homberger. Parallelization of a Two-Phase Meta-heuristic for Routing Problems with Time Windows. [J] *Asia-Pacific Journal of Operational Research* 2001(18): 35~47.
- [41] M.Haimovich and A.H.G.Rinnooy Kan. Bounds and heuristics for capacitated routing problems. [J] *Mathematics of Operations Research*, 1985(10):527~542.
- [42] W.Chiang and R.Russell. Simulated Annealing Metaheuristics For The Vehicle Routing Problem With Time Windows. [J] *Annals of Operations Research* .1996(63):3~27.
- [43] K.C.Tan, L.H. Lee and K.Q. Zhu. Heuristic Methods for Vehicle Routing Problem with Time Windows. [J]*Proceedings of the 6th International Symposium on Artificial Intelligence & Mathematics*, Ft. Lauderdale, Florida.2000.
- [44] H.Li, A.Lim, J.Huang. Local Search with Annealing-like Restarts to solve the VRPTW. [J] *European Journal of Operational Research* 2003, Vol.150,P115.
- [45] L.M.Gambardella, E. Taillard and G. Agazzi. MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. [J] *New Ideas in Optimization*,D. Come, M. Dorigo and F. Glover (eds), McGraw-Hill, London. 1999:63~76.
- [46] P.Kilby, P. Prosser and P. Shaw. Guided Local Search for the Problem with Time Windows. [J]*META-HEURISTICS Advances and Trends in Local Search Paradigms for Optimization*, S.Voss,S.Martello,I.H.Osman and C.Roucatirok(eds), Kluwer Academic Publishers, Boston., 1999, 473~486.
- [47] G.Kontoravdis and J.Bard, A GRASP For The Vehicle Routing Problem With Time Windows, [J]*Journal on Computing* 7, 10~23.

- [48] O.Braysy, and M.Gendreau, Evolutionary Algorithms for the Vehicle Routing Problem with Time Windows. [J] Working paper, SINTEF Applied Mathematics, Department of Optimization, Norway. 2002.
- [49] O.Braysy, and , G. Hasle and W. Dullaert. A Fast Local Search Algorithm for the Vehicle Routing Problem with Time Windows. [J]Working paper, SINTEF Applied Mathematics, Department of Optimization, Norway. 2002.
- [50] <http://web.cba.neu.edu/~msolomon/problems.htm>
- [51] Leong H W , Liu M. A multi-agent algorithm for vehicle routing problem with time window[J]. ACM 2006:106~111.
- [52] Alvarenga G B, Mateus G R. A two-phase genetic and set partitioning approach for the vehicle routing problem with time windows[J].Proc 4th International Conference on Hybrid Intelligent Systems,2004:428~433.
- [53] Andrew Lim, Fan Wang .A Smoothed Dynamic Tabu Search Embedded GRASP for m-VRPTW.[J] Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI2004) .
- [54] Alvarenga G B, Mateus G R. A two-phase genetic and set partitioning approach for the vehicle routing problem with time windows[J].Proc 4th International Conference on Hybrid Intelligent Systems,2004:428~433.
- [55] Oliveira, H. C. B. , Vasconcelos, G. C. , Alvarenga, G. B. Reducing traveled distance in the vehicle routing problem with time windows using a multi-start simulated annealing.[J] 2006 International Joint Conference on Neural Networks Sheraton Vancouver Wall Center Hotel, Vancouver ,BC,Canada 2006(7):16~21.
- [56] 李大卫, 王莉, 王梦光.一个求解带有时间窗口约束的车辆路径问题的启发式算法[J].系统工程, 1998,16(4):20~29.
- [57] 谢秉磊, 李军, 郭耀煌.有时间窗的非满载车辆调度问题的遗传算法[J].系统工程学报, 2000,15(3):290~294.
- [58] 宋厚冰, 蔡远利.有时间窗约束的车辆路径问题的改进遗传算法[[J].交通与计算机, 2003,21(4):25~27.
- [59] 刘小兰等.有时间窗的车辆路径问题的近似算法研究[[J].计算机集成制造系统, 2004,10(7):825 ~831.
- [60] 吴璟莉 李陶深. 遗传算法与禁忌搜索算法的混合策略在 VRPTW 问题上的应用[J].计算机工程与应用 2004(18):54~57.
- [61] 王德东 陈术山 郑丕谔. 确定车辆数的有时间窗车辆选径问题的混合算法[J].计算机应用,2006 26(2):482~484.
- [62] 占书芳 并行算法在带软时间窗车辆路径问题中的应用研究. 硕士学位论文.武汉: 武汉理工大学, 2006

致谢

光阴荏苒，两年多紧张而充实的研究生生活即将结束，在硕士论文完成之际，我要向所有支持、关心和帮助过我的人们表示最诚挚的谢意！

在研究生求学阶段，我师从于黄樟灿教授。黄老师的言传身教，为我树立了学习的榜样，不仅教会我做学问的方法，也教给了我做人的道理。黄老师科学严谨的治学态度和谆谆教诲、渊博的专业知识、严谨求实的治学态度和睿智宽容的学者风范、活跃开阔的学术思想，勇于探索的敬业精神，敏锐的洞察力、实事求是的作风、积极进取的科研精神是我学习的楷模；老师笑对人生、善待他人、至真至诚的生活态度将始终影响着我。更重要的是黄老师一丝不苟的治学风格，深深地引导我认真阅读大量文献，极大地丰富了自己的专业知识；同时每周一次的学术讨论，让我得以开阔眼界、更好的把握学术研究的动态。这些无时无刻不影响着我，将成为我终身受益的精神财富。

本文是我两年半来硕士期间学习成果的一次大总结。感谢所有指导过我的各位老师，他们在学术上和生活中都给过我无私的关怀，使我一点点地成长起来，在这两年多的人生路途中收获了很多。本文凝聚了导师大量心血，在论文选题、确定框架到撰写、修改全过程，得到导师悉心的指导，导师的意见和建议给予学生良多启发和帮助。

感谢我身边的同学和朋友们，给予我巨大帮助和支持，同时也感谢与我朝夕相处的室友，是她们在我最困难的时候，热心真诚地给予我帮助和鼓励！

感谢对我无微不至关爱的家人，他们是我前进的动力。

最后，衷心感谢在百忙之中评阅论文和参加答辩的各位专家、教授！

蒋文霞

2007/10/12 于武汉理工大学马房山

附录 1 实验数据 SolomonC101

CUST NO.	XCOORD D	YCOORD D	DEMAND D	READY TIME	DUE TIME	SERVICE TIME
1	40	50	0	0	1236	0
2	45	68	10	912	967	90
3	45	70	30	825	870	90
4	42	66	10	65	146	90
5	42	68	10	727	782	90
6	42	65	10	15	67	90
7	40	69	20	621	702	90
8	40	66	20	170	225	90
9	38	68	20	255	324	90
10	38	70	10	534	605	90
11	35	66	10	357	410	90
12	35	69	10	448	505	90
13	25	85	20	652	721	90
14	22	75	30	30	92	90
15	22	85	10	567	620	90
16	20	80	40	384	429	90
17	20	85	40	475	528	90
18	18	75	20	99	148	90
19	15	75	20	179	254	90
20	15	80	10	278	345	90
21	30	50	10	10	73	90
22	30	52	20	914	965	90
23	28	52	20	812	883	90
24	28	55	10	732	777	90
25	25	50	10	65	144	90
26	25	52	40	169	224	90
27	25	55	10	622	701	90
28	23	52	10	261	316	90
29	23	55	20	546	593	90
30	20	50	10	358	405	90
31	20	55	10	449	504	90
32	10	35	20	200	237	90

33	10	40	30	31	100	90
34	8	40	40	87	158	90
35	8	45	20	751	816	90
36	5	35	10	283	344	90
37	5	45	10	665	716	90
38	2	40	20	383	434	90
39	0	40	30	479	522	90
40	0	45	20	567	624	90
41	35	30	10	264	321	90
42	35	32	10	166	235	90
43	33	32	20	68	149	90
44	33	35	10	16	80	90
45	32	30	10	359	412	90
46	30	30	10	541	600	90
47	30	32	30	448	509	90
48	30	35	10	1054	1127	90
49	28	30	10	632	693	90
50	28	35	10	1001	1066	90
51	26	32	10	815	880	90
52	25	30	10	725	786	90
53	25	35	10	912	969	90
54	44	5	20	286	347	90
55	42	10	40	186	257	90
56	42	15	10	95	158	90
57	40	5	30	385	436	90
58	40	15	40	35	87	90
59	38	5	30	471	534	90
60	38	15	10	651	740	90
61	35	5	20	562	629	90
62	50	30	10	531	610	90
63	50	35	20	262	317	90
64	50	40	50	171	218	90
65	48	30	10	632	693	90
66	48	40	10	76	129	90
67	47	35	10	826	875	90
68	47	40	10	12	77	90
69	45	30	10	734	777	90
70	45	35	10	916	969	90

71	95	30	30	387	456	90
72	95	35	20	293	360	90
73	53	30	10	450	505	90
74	92	30	10	478	551	90
75	53	35	50	353	412	90
76	45	65	20	997	1068	90
77	90	35	10	203	260	90
78	88	30	10	574	643	90
79	88	35	20	109	170	90
80	87	30	10	668	731	90
81	85	25	10	769	820	90
82	85	35	30	47	124	90
83	75	55	20	369	420	90
84	72	55	10	265	338	90
85	70	58	20	458	523	90
86	68	60	30	555	612	90
87	66	55	10	173	238	90
88	65	55	20	85	144	90
89	65	60	30	645	708	90
90	63	58	10	737	802	90
91	60	55	10	20	84	90
92	60	60	10	836	889	90
93	67	85	20	368	441	90
94	65	85	40	475	518	90
95	65	82	10	285	336	90
96	62	80	30	196	239	90
97	60	80	10	95	156	90
98	60	85	30	561	622	90
99	58	75	20	30	84	90
100	55	80	10	743	820	90
101	55	85	20	647	726	90

附录 2 攻读硕士期间发表的论文

[1] 黄樟灿 蒋文霞 李书淦 有时间窗车辆路径问题的混合算法武汉理工大学学报 (信息工程版) 已录用