

## 摘 要

随着电子商务的飞速发展,物流业发生了新的重要变革,货物配送工作是其中一个重要的环节。在配送中心的货物配送工作中,配送方案优化问题是一个极具魅力的问题,优质的配送方案将产生极其可观的经济效益。

论文在分析配送中心配送方案优化方法的发展过程的基础上,研究了不同情况下的 VRP 问题的算法设计。根据顾客对配送时间的要求,分别建立了在不考虑配送中心库存成本的有软硬时间窗的单周期配送方案优化模型与在考虑配送中心库存成本的多周期配送方案优化模型。

在建模过程中分析了配送中心配送资源在不同情况下模型建立与求解方法。如果配送中心的资源有限,配送成本受顾客的库存影响,随机库存分配问题的解决决定着整个问题的解决。论文根据相关文献设计了用于解决这一问题的近似算法,将禁忌搜索算法用到了这一问题中,取得了较好的优化效果;如果配送资源不受限制,总配送成本不受顾客的库存成本的影响,问题转化为带有时间窗的车辆路径问题(VRPTW)。论文将这一问题分解为两个阶段来设计求解,在第一阶段使用遗传算法,第二阶段使用快速有效的节约启发式算法。

最后,在单车配送方案优化管理系统的开发中,解决了几种模型中都涉及的核心问题,即带有软硬时间窗的配送方案的确定。在开发过程中,由于顾客信息的不确定性,完成了建立动态基础数据库。在配送方案优化过程中,充分考虑了顾客对软硬时间窗的要求,最终生成了车辆的最优配送方案一维显示与成本计算,能基本满足实际应用。

**关键词:** 配送方案优化; 时间窗; 库存; VRPTW

---

---

## ABSTRACT

Logistics trade has a newly important change, and the freight distribution is a important link with the rapid development of electronic business. In the course of freight distribution in a distribution center, the distribution plan optimization problem is a fascinating one. A reasonable distribution plan can bring extremely considerable economic benefit.

In the base of analyzing the development process of distribution plan optimization in a distribution center, we research the algorithm design of VRP under different conditions in the thesis and set up the distribution plan optimization matrix including considering the single cycle in that inventory cost is not involved with soft and hard time windows and more cycles in that inventory cost is involved apart according to the customs' desire to the distribution time.

We analyze the method of setting up matrix and the solution in the different conditions. If the resource in the distribution center is finite, the custom inventory affects the distribution cost and the stochastic inventory allocation problem is the key to whole problem's resolving. A approximate algorithm is designed in order to resolve this problem according to interrelated documents in this thesis in that the Tabu Search is used and a better optimization result is obtained; If the resource is infinite in the distribution center, the distribution cost is not affected by the custom inventory, the problem is a vehicle routing problem with time window (VRPTW). Then the problem is disintegrated into two parts that are solved respectively. Genetic algorithm is used in the first part and the saved heuristics algorithm is used in the second part.

In the end, in the course of the management system development about single vehicle distribution plan optimization the key problem has been solved in the matrixes concerned that are the determination of the distribution plan with soft and hard time windows. In process of development, the dynamic basis database has been set up under the condition of the uncertainty about the custom's information. After abundant consideration to the customs' require to soft and hard time window, the vehicle optimized distribution plan has been

---

formed with is a one-dimensional show and the total cost is computed at the same time. All of these can satisfy practical appliance.

**Keywords:** distribution plan optimization; time window; inventory; VRPTW

---

## 第 1 章 绪论

物流作为“第三利润源泉”对现代的经济活动影响日益明显，越来越引起人们的重视。现代物流业已经成为把握竞争优势的有效方式，将为国民经济在高起点上持续发展，提供基础动力。在经济全球化和信息化的推动下，现代物流业已从为社会提供传统运输服务，扩宽到以现代科技、管理和信息技术为支柱的综合物流系统。

配送是物流中一个重要的直接与消费者相连的环节，是货物从物流结点送达收货人的过程。配送是在集货、配货基础上，按货物种类、品种搭配、数量、时间等要求所进行的运送，是“配”和“送”的有机结合，配送的流程一般如图 1-1 所示。在传统的配送系统中，由于商品的需求量及种类较少，零售商可凭借较多的存货及较长的订货周期来减少供货商的配送频率，以降低运输成本。但是在现代的配送系统中，零售商为了减少资金积压及提供多样化的商品，势必要减少各种商品的存货数量，而同时又必须考虑到提供最好的服务品质(不允许缺货)。这就要求配送中心对商品的仓储与运输进行有效合理的统筹规划以降低配送成本。

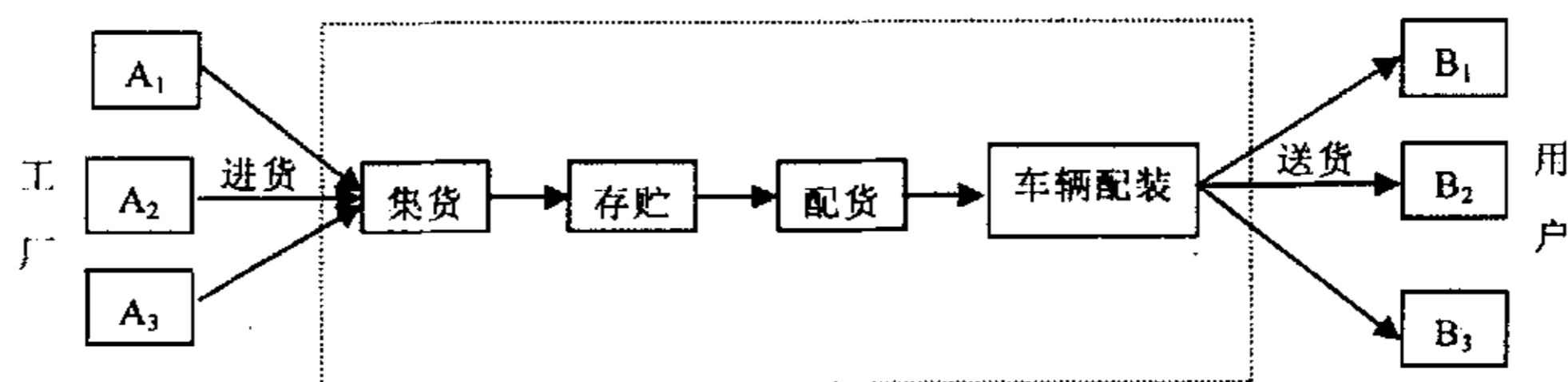


图 1-1 配送流程图

### 1.1 论文研究的背景及意义

配送过程主要包括从生产工厂进货并集结的集货作业；根据各个顾客的不同需求，在配送中心将所需要的货物挑选出来的配货作业；考虑配送货物的质量和体积、充分利用车辆的载重和容积的车载货物的配装及配送路线的配送方案的确定。而在整个过程中，配送方案是非常重要的一个问题，配送方案是指配送中心按不同客户的多频度、小批量的订货要求组织配送，其主要内容是指根据顾客的货物需求量来分配车辆和选择优化路线，即如何将车

辆有效的使用并决定其最经济的行驶路线，使商品能在最佳的时间内送到顾客的手中。这类问题是典型的车辆路径问题，即 Vehicle Routing Problem，简称 VRP 问题。

由于从事配送的车辆尤其是从事城市配送的车辆工作条件复杂，不仅货运点多、货物种类多，而且运输服务地区内运输网点分布也不均匀，同时，很多客户还对配送提出了时间要求，因此，如何应用现代数学方法及计算机快速求解优化配送方案是国内外专家学者普遍探索的重要课题。

配送方案优化问题同时也是供应链研究的一项重要内容。在配送作业中，管理者们需要采取有效的配送方案以提高服务水平，降低货运费用，配送车线路是否能合理对配送的速度、成本、效益有着直接的影响。选取恰当的配送方案，可以加快对客户需求的响应速度，提高服务质量，增强客户对物流环节的满意度，降低服务商运作成本。

本论文从为配送中心提供一种具有一定普遍性的配送方案优化方法的目标出发，在综合考虑顾客需求量、到货时间以及库存量的情况下对一对一配送和一对多配送的配送方案优化问题进行分析，并建立配送中心配送成本最小的目标优化模型和设计求解算法。在此基础上，研究如何用计算机实现具有一定普遍性的带有软硬时间窗的单车配送方案优化的模拟过程，这一过程的实现对配送企业实现计算机配送调度，降低成本和提高物流经营管理水平将具有重要的参考价值，同时由于这种模拟带有很强的通用性，所以对于一般物流企业的配送方案优化过程具有一定的普遍性及适应性。

## 1.2 配送方案优化问题介绍

### 1.2.1 问题的提出

配送方案优化问题是一个极具魅力的优化问题，在 1959 年由 Dantzig 和 Ramser 以配送路径优化的问题首次提出后，相关的配送方案优化问题就一直吸引着全世界无数的科学家、工程师和管理者为之探索。这不仅因为它是一类求解较难的组合优化问题，是对人类智慧的考验，而更重要的是因为它有很强使用背景，可产生极其可观的经济效益。

物流配送的类型和形式多种多样，但大部分配送形式中的运输特点基本相同，即将商品从配送中心按一定要求发送到多个需求点。配送方案(如车辆的分配、路径的选抒、配送的准时与快捷等)制订的优化与否，对提高配送效

---



率及服务水平、降低货运费用等具有极其重大的意义。

然而实际配送时，需求点较多且分布不均，道路网又比较复杂，单凭人工经验已经难以制定令人满意的配送方案，随着物流业的不断规范，现在的顾客对配送服务质量要求不断提高，对配送的时效性要求较强，在规定时间内要求配送的货物原则上一定要配送到位，在不能按时到位的情况下，会对配送中心的成本产生负影响，同时顾客的需求量与库存同样影响着配送中心的作业成本，如何在保证高度准时、快速配送和满足顾客要求的情况下降低成本是配送中心配送方案优化过程中所要面临的一个重要课题。所以，设计适合的配送方案优化模型和算法以及应用计算机技术实现模拟优化过程就是当前所面临的一个重要的课题。本论文就是在这种情况下，着重分析了配送中心本着成本与顾客需求量、时间和库存三者之间的关系，改进了相应的数学模型和设计了解算法，最后并实现了方案优化过程中关键技术的计算机模拟过程。

### 1.2.2 配送方案优化问题在配送中的应用

在多数发达国家，物流业一直都是经济中一个重要的组成部分，而在发展中国家，物流业对经济的影响也越来越重大；对于个体公司来讲，物流成本是运营成本中的一个重要部分。这种经济的重要性以及在社会上应用的广泛性都激发利益追求者和科学界研究人员去从事运用运筹学与管理科学去提高物流业中配送效率的研究。这些研究工作都不断推动着配送方案优化方法的不断应用与进步。在运输科学中最早有关配送方案优化问题的一个重要的领域就是车辆路线问题(VRP)，VRP 问题在物质分派与物流学中占有着重要的作用。

目前，传统 VRP 问题的形式已有很大的发展，该问题已不仅局限于汽车运输领域，在水运、航空、通讯、电力、工业管理、计算机应用等领域也有一定的应用，同时，随着物流业的发展以及对顾客的需求越来越重视的情况下，改进的 VRP 问题也在配送中起着重要的作用。有关配送方案优化的算法已用于航空乘务员轮班安排、轮船公司运送货物经过港口与货物安排的优化设计、交通车线路安排、生产系统中的计划与控制等多种组合优化问题。

在解决以下实际问题中配送方案优化同样也起着重要的作用：

1. 学校汽车路线学校车队需要在早上接学生上学，下午送学生回家。学校汽车路线问题包括用学校的变动数量的车队从事先预定的位置为学校运送学

生，目标就是除了使汽车的走行距离最短和学生的旅行时间最短外在满足所有学生的要求后使所用的汽车数量最小。

2. 包裹运送与收取像 UPS 和包裹服务公司和联邦快递喜欢为司机高效决定配送和/或收货路线。这里，必须考虑产品的优先性。如，需要两天到达的航空包裹必须于次日 1000 点之前被送到。也必须在车辆路线问题中把地域性考虑进去。

3. 软饮料的配送如可口可乐这样的软饮料公司喜欢向超市或地方便利店配送他们的产品的时候选择经济型路线。

4. 银行收款一天的商务结束，中心银行要从他的分行收取存款。为安全起见，一辆车所收款额数量不能超过一定的数量。

5. 垃圾回收垃圾处理公司需要安排垃圾车为城市地区服务。对于居民区，由于需要覆盖每一条街道，这种问题实际上就是一个弧形路线问题。

6. 食品配送把准备好的食品配送给那些不能亲自为自己购买或为自己制作的顾客。在这一问题上，顾客是随机的，并且顾客要求食品必须在一定时间内送达到顾客手中。

7. 牛奶业从农场收集牛奶运输给处理中心。

8. 服务业一个顾客集对诸如修护人员、邮递员或护士的上门服务的需要。

### 1.3 国内外研究现状

配送方案优化问题最早是由 Dantzig 和 Ramser 于 1959 年以配送路径优化的问题首次提出，对于配送路径优化问题的一般定义为对一系列装货点和(或)卸货点，组织适当的行车线路，使车辆有序地通过它们，在满足一定的约束条件(如货物需求量、发送量、交发货时间、车辆容量限制、行驶里程限制、时间限制等)下，达到一定的目标(如路程最短、费用最少、时间尽量少、使用车辆数尽量少等)。

国外对配送方案优化的相关问题作了大量而深入的研究，例如 1983 年 Bodin、Golden 等人在他们的综述文章中就列举了 700 余篇文献。在 Christofides(1995)，Golden 和 Assad(1998)编辑的论文集，以及 Altinkemer 和 Gavish(1991)，Laporte(1992)，Aalhi(1993)等的综述文章中都进行了详尽的阐述。该领域的代表人物有 Bodin, Christofides, Golden, Assad, Ball, Laporte, Rinnooy Kan, Lenstra, Desrosiers 和 Desrochers 等人。

### 1.3.1 配送方案优化问题研究现状

目前对于配送方案优化问题的研究大多致力于改进 VRP 问题,而大多数的配送调度问题可以归结为配送车辆的分配、行车路线的组织问题(VRP),即根据一定的要求,(即目标函数,例如运距最短、配送时间最短、运输费用最少等),将配送过程归结为表述问题的数学模型,然后用计算机求得合理可行的优化方案,在生产中付诸实施。传统 VRP 问题以及各类改进的 VRP 问题是组合优化领域中著名的 NP 难题。近二十年来,无论在国内外,VRP 问题都是一个非常活跃的研究领域。目前用于解决该问题的现代数学方法主要分为以下几类

1. 精确优化方法。包括制定下界及相关分枝定界算法(Branch and bound approach)(算法比较适用于求解小型问题)、切平面法(Cutting planes approach)(算法考虑所有可能的可行解子集,在此基础上进行重复优化求解,其算法本质是最短路径算法结合分枝定界法)、动态规划法(Dynamic Programming approach)(算法针对固定车辆数的 VRP,通过递归方法求解。为减小问题的计算规模,引入可行性规划或松弛过程减少状态的数量。算法的核心思想是最佳化原理,运用数学逻辑来处理一连串相互关联的决策问题,并采取系统优化的步骤以求得对整体有利的方案)、网络流算法(Network flow approach)(该方法结合图论中的一些传统算法(如最短路径、宽度搜索算法等),经常可以解决一些搜索与动态规划无法解决的非 NP 问题。运用该方法解决具体问题时,没有现成的模式可以套用,需要根据网络流的性质发挥创造性探索建立多种模型,通过分析、选择、优化、确定适当的模型)与三下标车辆流方程(针对带能力约束、时间窗口以及无停留时间的 VRP 问题而提出。该方法在模型中有效引入了代表时间窗口的变量,从而可适用于通用任务分配问题(GAP)和带时间窗口的 TSP)等。但精确算法的计算复杂度很高,随着运输系统的复杂化和对调度的多目标要求,获得整个系统的精确优化解越来越困难,花费的时间和费用太大,因此精确解法不能应用于规模较大的配送车路由问题的求解,仅用于运输调度的局部优化问题。

2. 启发式方法(Heuristics)。指通过经验法则来求取运输过程满意解的数学方法。启发式方法能同时满足详细描绘问题和求解的需要,较精确优化方法更为简单实用,缺点是难于知道什么时候好的启发式解已经被求得。启发式方法中最具代表性的就是由 Clarck 和 Wright<sup>[15]</sup>提出的节约法(Saving Method),算法的基本原理是考虑约束限制(如容量等),按节约值由大到小排



序, 依次将对应的两顾客点排入路径中, 直到所有顾客都被排入路径为止。许多成功的车辆调度软件就是根据该方法或其改进方法开发的。西南交大的李军<sup>[1]</sup>针对有时间窗的车辆路线安排问题提出了一种利用节约法的启发式算法。典型启发式算法中还包括由 Lin 和 Kernighan<sup>[20]</sup>提出的, 并由 Christofides<sup>[21]</sup>和 Gilbert LaPorte<sup>[22]</sup>等人所推广的分支交换探索法, 该算法始终保持解的可行性而又略图向最优目标前进。在每一步, 都改变一个可行解而减少总费用, 直到这个过程继续到不再可能使费用减少为止。Gillet 和 Miller<sup>[17]</sup>提出的扫描法(Sweep Method)先把节点或弧的需求进行分组或划群, 然后对每一组按旅行商问题(TSP)求解, 设计出一条经济的路线。此外, 常用的启发式方法还有 2-opt 法、不完全树搜索算法等。各种启发式方法的主要区别在于收敛的速度和程度不同。

3. 模拟方法(Simulation)。模拟方法于 1986 年由 Golden 和 Skiscim 最早提出<sup>[18]</sup>, 此方法通常是利用常数公式、逻辑表达式、图表、坐标图形等抽象概念表示实际运输系统内部状态和输入输出的关系, 以便通过计算机对模型进行实验, 通过实验取得改善运输系统或设计新运输系统所需信息。虽然模拟方法在模型构造、程序调试、数据整理方面工作量大, 但由于运输系统结构复杂, 不确定情形多, 模拟方法仍以其描述求解问题的能力优势, 成为复杂运输调度系统建模的主要方法。这种方法主要应用于行车路线与物流中心区位的选择, 优点在于可直接观察系统安排的效率和效果, 但由于问题的实际情况多变且具不确定性, 是否能将实际的配送情形系统逻辑化为仿真程序, 往往令人质疑。

4. 交互式优化法。该方法结合人类决策与计算机计算能力, 在求解过程中, 通过高度的人机交互模式, 结合专家的决策信息, 并据以计算出结果; 优点是寻优的过程中, 决策者可以很清楚地看到各约束条件之间的替代关系, 以及参数变化可能导致的成本变化。缺点在于由于在模型的优化过程中加入了人的主观因素, 所以通常会增加模型最终实现并被采用的可能性。采用这种方法求解车辆路径问题的早期雏形是由 Krolab, FeltS, Marble 和 Nelson<sup>[19]</sup>等人提出的。

精确优化方法是以往广泛采用的方法, 另外三种方法则代表了最近研究方向。尤其是启发式方法, 作为一种逐次逼近的算法, 虽然不一定得到最优解, 但是可以高效率地得到具有较高精度的解, 而且也易于考虑各种实际问题, 因此, 现已成为解决配送问题的重要方法。与传统的启发式方法相比,

近年来一些新的启发式方法(元启发式方法),通过对启发式规则和搜索方式的改进,在求解多节点、多约束的 VRP 问题上可以获得较快的收敛速度和较高质量的全局解,从而进一步拓宽了启发式方法的应用领域。主要的元启发式方法有遗传算法、禁忌搜索算法、模拟退火算法和神经网络算法等,在论文后有关的参考文献中对于如何运用遗传算法来解决车辆调度问题作了详细的阐述,对于运用禁忌搜索算法来探讨此问题的解决方案,均取得了良好的效果。

由上述内容可知,对于配送方案的制订,一般对 VRP 问题、带有时间窗的 VRP 问题或在考虑顾客库存情况下的 VRP 问题较多,而对于把顾客需求量、时间以及库存三者结合起来考虑并建立模型的研究还处于刚刚起步阶段。

### 1.3.2 配送方案计算机管理系统开发现状

近年来,随着计算机在性能上的迅速发展,国外应用计算机及现代数学方法制订配送方案以优化运输管理效果的研究工作,也在较大范围内快速展开。在配送方案的形式、构造、分析以及求解方法的实现上,都有了许多突破。某些常用且较成熟的算法已被人们运用于实际配送方案制订系统。据相关资料显示,美国利用最短路算法、启发式算法开发出计算机配送调度系统可用来解决货运汽车作业计划中路线选择和车辆分配等问题,使汽车里程利用率提高 5~15%,运输成本和运输时间也有了明显下降。在国外开发的配送方案调度系统中,主要包括由美国 IBM 开发的 VSPX 系统,其核心算法是最短路径算法和启发式算法;由日本富士通公司开发的 VSS 系统,其核心算法是节约法;由美国美孚公司开发的 HPCAD 系统,其核心算法是扫描法。

迄今为止,有关国外研究配送调度理论及算法的文献很多,并且在计算机调度应用方面也取得了一定的成果,但是,国内在这方面的研究大多数停留在理论层次上,实际应用系统的开发才刚刚起步。随着配送作业的兴起,对车辆调度提出了经济性、准时性、灵活性的综合要求。虽然理论界和物流企业界早在数年前就提出建立配送中心信息系统,但计算机配送优化调度系统开发上还处于刚刚起步阶段。主要是由于配送业在我国还未真正开展,企业对配送优化方案认识不够;配送方案优化数学模型本身与配送应用结合上也存在一些问题,大多数算法设置了许多假设条件,且仅考虑了 VRP 问题的某些约束,而实际配送时顾客对运输的要求较高,调度约束条件多,这样就限制了算法的应用范围。

## 1.4 本论文的主要内容和路线

### 1.4.1 本论文研究内容

本论文通过对配送中心配送方案优化方法分析,根据传统 VRP 问题及改进 VRP 问题的求解方法,对解决配送过程中在考虑顾客需求量、时间窗和库存量三种情况下的有关单周期单货物品种配送和多周期多货物品种配送的配送方案优化过程建立了不同的数学模型,对模型设计了求解算法,最后对带有软硬时间窗的单车配送方案优化过程实现了计算机模拟过程,本论文的主要内容如下:

第一章 绪论,主要介绍国内外研究背景及意义;配送方案优化研究在现代配送中的应用以及配送中心配送方案优化问题的提出;

第二章配送方案优化方法分析,主要分析了国内外近年来用于解决传统 VRP 问题以及改进的 VRP 问题的主要历程及其中所用到的主要算法;

第三章数学模型的建立与求解,主要研究在不考虑配送中心库存成本的单周期配送方案优化过程与考虑配送中心库存成本的多周期配送方案优化过程中的数学模型的建立与算法设计。在单周期配送方案制订过程中,着重研究了在考虑配送中心的配送资源有限与无限两种情况下的模型建立与求解;

第四章 配送方案优化的计算机实现,针对配送中心配送方案制订过程中,带有软硬时间窗的单车配送方案的制订开发了配送方案优化管理系统,主要研究了在顾客之间信息不确定的情况下,如何生成并调用动态基础数据,并最终显示生成的优化配送方案以及总配送成本;

结论,总结整个论文所得到的结论以及有待进一步研究的问题。

### 1.4.2 本论文的思想方法和技术路线

1. 从系统分析的思想出发,研究了配送方案优化的发展历程,分析现代配送中心在进行配送工作的时候所面临的主要任务与挑战。
  2. 研究配送中心配送过程中的配送成本与顾客需求量、到货时间和库存三者之间的关系,对配送方案优化方法做出科学的结论。
  3. 科学分析相关算法的理论基础及应用现状,研究配送方案优化过程中不同情况下数学模型的建立以及相应的求解算法的设计和求解过程;
  4. 研究数学算法的实现机理以及实际应用的可能性,研究配送过程中带有软硬时间窗的单车配送方案优化过程的计算机模拟。
-

## 第 2 章 配送方案优化方法分析

从传统 VRP 问题提出至今,所有与 VRP 相关的问题,其最终目的都是制订出可行的配送方案以实现对经济利益最大化的追求。本章将介绍国内外学者对配送方案制订的发展历程及相关的主要模型与算法。配送方案的制订从开始出现至今,从简单的单一车辆的路径优化研究一直发展到今天的对综合考虑顾客的需求与库存量的研究。本章只介绍传统 VRP 问题、含时间窗的 VRP(VRPTW)问题和考虑顾客库存的 VRP 问题。

由于第三章中的涉及的模型中,经分解后用到了旅行商问题(TSP)的求解算法,所以在这里首先对 TSP 问题进行描述。

### 2.1 旅行商问题描述

旅行商问题(Traveling Salesman Problem 简称 TSP)是 VRP 问题的一个特例(即当 VRP 问题中只包括一条路径,且没有能力约束的时候,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,引起了人们的极大兴趣,现已归入 NP-完备问题类。TSP 本身可以直接用于解决类似 TSP 的最优巡回线路等问题。

TSP 一般描述为旅行商从驻地出发,经所要去的城市至少一次后返回原地,应如何安排其旅行线路,才能使总的旅行距离(或时间、费用等)最少。对于现实问题,由于限制条件的增多,TSP 衍生出许多相关问题。图 2-1 显示了 10 个城市的 TSP 问题的 181440 个解中的一个。

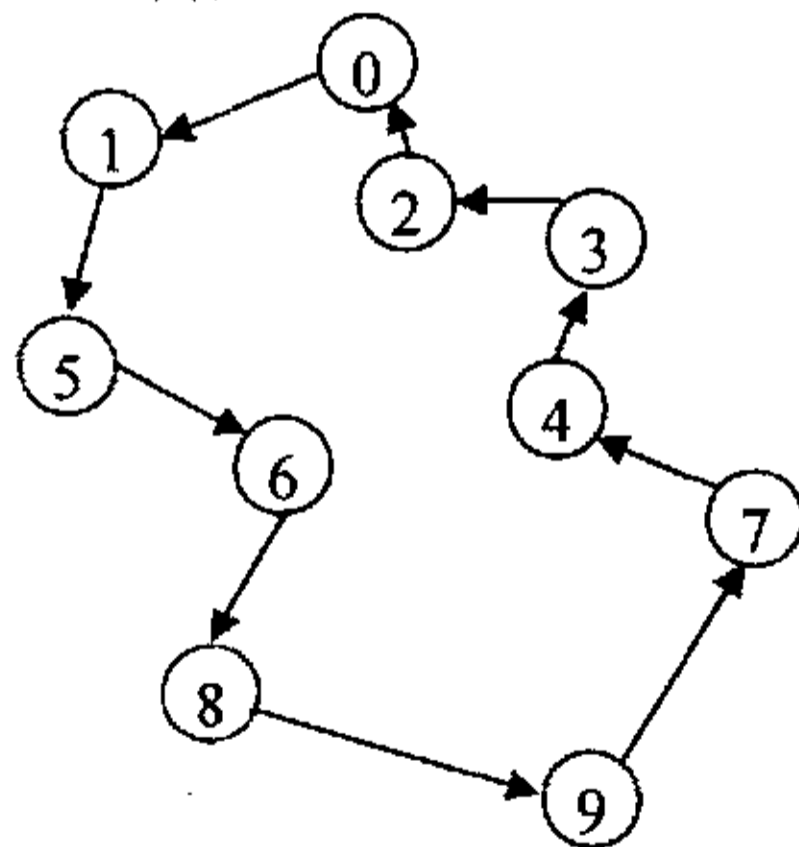


图 2-1 10 城市 TSP 问题的一组解



TSP 问题的解决方法可以通过启发式算法的多样的改变而获得。最基本的三种启发式算法的主要方法可以予以区别

- 精确运算启发式算法
- 局部搜索启发式算法
- 后启发式算法

#### 1、精确运算启发式算法

TSP 问题整型规划公式表达是在有  $n$  个基点的完全有向图  $G=(V, A)$  中定义的, 基点集合  $V=\{1, 2, \dots, n\}$ , 弧线集合  $A=\{(i, j)|i, j=1, 2, \dots, n\}$  与弧相关联的非负成本为  $c_{ij}$ 。这样, TSP 问题数学表达如下, 同 Dantzig, Fuleerson 和 Johnson(Lawler, 1985)的表达

$$\text{目标 } \min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (2-1)$$

$$\text{s.t. } \sum_{j \in V} x_{ij} = 1 \quad i \in V \quad (2-2)$$

$$\sum_{i \in V} x_{ij} = 1 \quad j \in V \quad (2-3)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \text{for all } S \in V, S \neq \emptyset, \quad (2-4)$$

$$x_{ij} = 0 \text{ or } 1, \quad i, j \in V \quad (2-5)$$

这里, 如果弧  $(i, j)$  属于解中的部分,  $x_{ij}=1$ ; 否则  $x_{ij}=0$ 。

在这个公式中, 目标函数(2-1)描述了最优路线的成本。约束条件(2-2)和(2-3)是次数约束。约束(2-2)说明每一个顶点正好进入一次, 约束(2-3)说明每个顶点也正好出去一次。约束(2-4)是次旅行线排除约束。这些约束禁止了次旅行的形成, 即, 在少于  $n$  个点的子集合上的旅行, 这里  $|S|$  是子集合  $S$  的基数次序。最后, 约束(2-5)说明了变量的二重性。

分枝定界法通常被用于解决 TSP 问题。在数学规划的文章中, 这种方法可以被看作是一些问题的初始松弛约束, 然后通过列举的步骤重新获得可行解。分枝定界法的质量直接与松弛度所提供的界线的质量相关。对于 TSP 问题, 一个初始较低的界线可以通过由上面松弛约束(2-4)所描述的公式表达而得到。这样问题就成为分配问题。许多作者对 TSP 问题提出了分枝定界法。以分配问题的  $x$  松弛为基础, 最近用分枝定界法的研究者是 Lysgaard(1999)。



Fischetti(1997)、De Kort(1993)、Malandraki(1993)和 Sarkar(1991)。

## 2、局部搜索启发式算法

最近的邻居法、Clarke and Wright 节约法、最近插入法和最省插入法是求解 TSP 问题的最著名的以贪婪为基础的结构化启发式算法中的一部分。这些都是通过数据在单一搜索中建立一个解的单一路径的启发式算法。这些启发式算法在贪婪原则的基础上都已经得以改进，他们包括以下三种特有特性 (Murty, 1995)<sup>[10]</sup>

(1)增加特性。以贪婪为基础的启发式算法以下面这种方法说明问题解可以被看作或是元素集的子集、或者是在某种次序下元素集的序列。这种方法建立了一个解集，或是一个解的序列，每次从零开始的一个元素，并且以第一个完全解而终止。

(2)无回路特性。一旦选择了一个元素作为解集中所包含的一个元素，这个元素就永远不能取消或是被其它元素代替。也就是说，在贪婪算法中，在算法中某一步骤决定是永远不能在随后被修改的。

(3)贪婪选择特性。当通过这一准则而考虑时，对解集中所包含的每一个新的元素的选取，或是对在序列中下一个位置的填充的元素选取，是根据一定准则在那一步骤中所能选取的元素中最好的一个，也就是说，每一个新元素在那一步骤中都对总成本贡献着最少量的成本，或是对总收入而言贡献最大的量。

当与以  $K$ -opt 和/或  $Or$ -opt 准则为基础的改进启发式算法一起结合使用的时候被称为局部搜索启发式算法或下降方法。“这些局部搜索启发式算法的搜索策略是盲目的。这就意味着借此可以产生新解的次序仅仅依靠在启发式算法执行过程中所收集到的信息。如果目标函数值不可能进一步的改进了，启发式算法终止，并导致了陷入局部最优，并使算法在局部最优中下降。(Breedam, 2001)”。用这些启发式算法的部分早期研究者有 Bodin(1981)、Lenstra(1976)、Lin(1973&1965)、Clarke(1962)和 Croes(1958)。

## 3、后启发式算法

后启发式算法与局部搜索启发式算法不同，它可以通过临时接受能够产生不利的目标函数值的移动而成功的跳出局部最优。在后启发式算法中被广泛使用的是遗传算法、模拟退火法和禁忌搜索算法。

### (1)模拟退火算法(simulatedannealingalgorithm)

这是一种源于五十年代、基于 MonteCarlo 迭代求解思想的随机搜索算

---

法,八十年代才开始应用于组合优化领域,其出发点是将组合优化问题与统计力学的热平衡作类比,把优化的目标函数视作能量函数,模仿物理学中固体物质的退火处理,先加热使之具有足够高的能量,然后再逐渐降温,其内部能量也相应下降,在热平衡条件下,物体内部处于不同状态的概率服从 Boltzman 分布,若退火步骤恰当,最终会形成最低能量的基态。这种算法思想在求解优化问题时,不但接受对目标函数(能量函数)有改进的状态,还以某种概率接受使目标函数恶化的状态,从而可使之避免过早收敛到某个局部极值点,也正是这种概率性的扰动能够使之跳出局部极值点,故而得到的解常常很好。经典的模拟退火算法流程为

Step1 选择初始状态  $H$ (初始解)、初始温度、降温次数;

Step2 生成  $H$  的邻域状态  $H'$ , 并计算  $Z(H')-Z(H)$ ;

Step3 按接受概率置换  $H$ ;

Step4 重复 Step2 直至停机条件满足。

模拟退火法所得解的好坏与初始状态、温度函数等都有一定的联系,降温较快的效果不一定很好,效果好的,其降温过程又极其缓慢。但由于该方法适用范围广,并可人为控制迭代次数,反复求解,因此具有很强的实用性。

## (2)遗传算法(geneticalgorithm)

这是一种(或者说一类)来自生物进化理论中“自然选择、适者生存”原则的搜索(寻优)算法,它基于生物学的自然选择原理和自然遗传机制模拟生命的进化,这种新近发展起来的完全异于传统思想的搜索和优化方法自 JohnHolland 等(1975)提出以来,获得了广泛的应用。尤其是在人工智能领域,取得了极大的成功,对于许多复杂困难问题的解决提供了强有力的处理办法。

遗传算法用于优化问题时,其最主要特征就是它不在单点上寻优,而是从整个种群(Population)中选择生命力强的个体产生新的种群;它使用随机转换原理而不是确定性规则来工作.遗传算法中常用的遗传操作包括三个基本算子:繁殖(Reproduction)、交叉(Crossover)、变异(Mutation)。其中,繁殖算子用于从旧种群生成新种群,交叉算子用于从父母代生成子代,变异算子用于对子代作某种变异。这里,繁殖和交叉操作是遗传算法的有效性所在,但有时会丢失一些重要的遗传信息,适当的变异操作可保证这些有用信息的引入。遗传算法在具体实施中有多种变形和修正,其主要操作思想可描述成:

Step1 问题的染色体表示;

Step2 初始解组(种群)的生成;

- Step3 计算解组中各个解的适值函数(代价函数);
- Step4 从解组中随机抽取两个解作为父母代;
- Step5 对父母代实施遗传操作(交叉、变异等)以产生一个后代解;
- Step6 按某种规则,用该后代解替换原解组中的某个解;
- Step7 若当前解组符合停机条件,则算法终止,否则,转 Step4。

遗传算法所得结果的好坏,主要依赖于遗传代数和解组规模,在实际应用中只能根据具体要求,在合理的时间内对问题进行求解,若所得解不能令人满意,可增大解组规模或遗传代数,从而有希望得到问题的全局最优解,当然,这是以延长计算时间为代价的。

### (3)其它算法

除了前面提及的一些典型的 TSP 算法之外,还有其它许多方法可供使用,如, Norback 等人(1977)的几何图解法、我国学者提出的极小代数算法(1990)、归约算法(1991)、从模拟退火法引申而来的均场退火法、遗传退火法、源于神经网络并本质上作为并行实现的弹性网法、自组织映射法,等等。还有一种使用效果也相当不错的 Tabu 搜索法,它采用了类似爬山法的移动原理,将最近若干步内所得到的解储存在 Tabu 列表中,从而强制搜索避免再次重复表中的解,该方法在九十年代曾求解过几十万个顶点的大型 TSP。

## 2.2 传统 VRP 问题

### 2.2.1 传统 VRP 问题描述

传统 VRP(Vehicle routing problem)可以表示为给定一个中心仓库(central depot)一个车辆集合和一个顾客集合,车辆和顾客各有自己的属性,每辆车都有车的容量,所装载的货物不能超过它的容量。起初车辆都在中心点,顾客在空间任意分布,车把货物从车库运送到每一个顾客(或从每个顾客处把货物运到车库),要求满足顾客的需求,车辆最后返回车库,每个顾客只能被服务一次,怎样才能使运输费用最小。

### 2.2.2 传统 VRP 问题模型

#### 1. 问题记号

$n$ =顾客的数目,顾客标号从 1 到  $n$ ,标号 0 表示配送中心; $Q_k$ =车辆  $k$  的容量; $q_i$ =顾客  $i$  的需求; $c_{ij}$ 从顾客  $i$  到顾客  $j$  的旅行费用; $M_k$ =使用车辆  $k$  所需付出的费用;

---

$$\text{变量 } y_{ik} = \begin{cases} 1 & \text{若服务顾客 } i \\ 0 & \text{其他} \end{cases}; \quad x_{ijk} = \begin{cases} 1 & \text{若车辆 } k \text{ 顾客 } i \text{ 直接旅行到顾客} \\ 0 & \text{其他} \end{cases}$$

## 2. 数学模型

$$\min \sum_{i,j,k} (c_{ij} x_{ijk} + M_k) \quad (2-6)$$

$$s.t. \sum q_i y_{ik} \leq Q_k \quad k=1,2,\dots \quad (2-7)$$

$$\sum_{ik} y_{ik} = \begin{cases} \text{使用车辆的数目} & i=0 \\ 1 & i=1,\dots,n \end{cases} \quad (2-8)$$

$$y_{ik} = 0 \text{ 或 } 1 \quad i=0, \dots, n, k=1, 2, \dots \quad (2-9)$$

$$\sum_i x_{ijk} = y_{jk} \quad j=0, \dots, n \quad k=1, 2, \dots \quad (2-10)$$

$$\sum_i x_{ijk} = y_{ik} \quad i=0, 1, \dots, n \quad k=1, 2, \dots \quad (2-11)$$

$$\sum_{i,j \in S, k \in S} x_{ijk} \leq |S| - 1 \quad S \subseteq \{1, \dots, n\}, k=1, 2, \dots \quad (2-12)$$

$$x_{ijk} = 0 \text{ 或 } 1 \quad i, j=0, 1, \dots, n, k=1, 2, \dots \quad (2-13)$$

束缚(2-7)-(2-9)保证了每条路线起始于并最终返回中心车库 0 点, 每个顾客都被访问一次且只被访问一次, 车的装载量不超过车的容量, (2-10)-(2-13)定义了对于一个给定车辆的旅行商问题。

### 2.2.3 传统 VRP 问题算法分析

#### 2.2.3.1 启发式算法

##### 1. 启发式算法理论

启发式源自英文单词 heuristics, 启发式方法意为通过对过去经验的归纳推理以及实验分析来解决问题的方法, 即借助于某种直观推断或试探的方法。启发式方法要求分析人员必须运用自己的感知和洞察力, 从与研究问题有关而且比较基本的模型及算法中寻求期间的联系, 从中得到启发, 去发现适于解决该问题的思路和路径。

用启发式方法解决问题时强调“满意”, 常常是行到满意解, 决策者就认为

可以了，而不去追求最优性和探求最优解。

## 2. 启发式方法的求解过程

用启发式方法求解问题是通过不断的迭代过程实现的，因而需拟定出一套解的搜索规则。为能得到满意，在整个迭代过程中要不断吸收新的信息，必要时改变原来拟定的不合适的策略，建立新的搜索规划，注意从失败中吸取教训，并逐步缩小搜索范围。具体求解如下图 2-2。

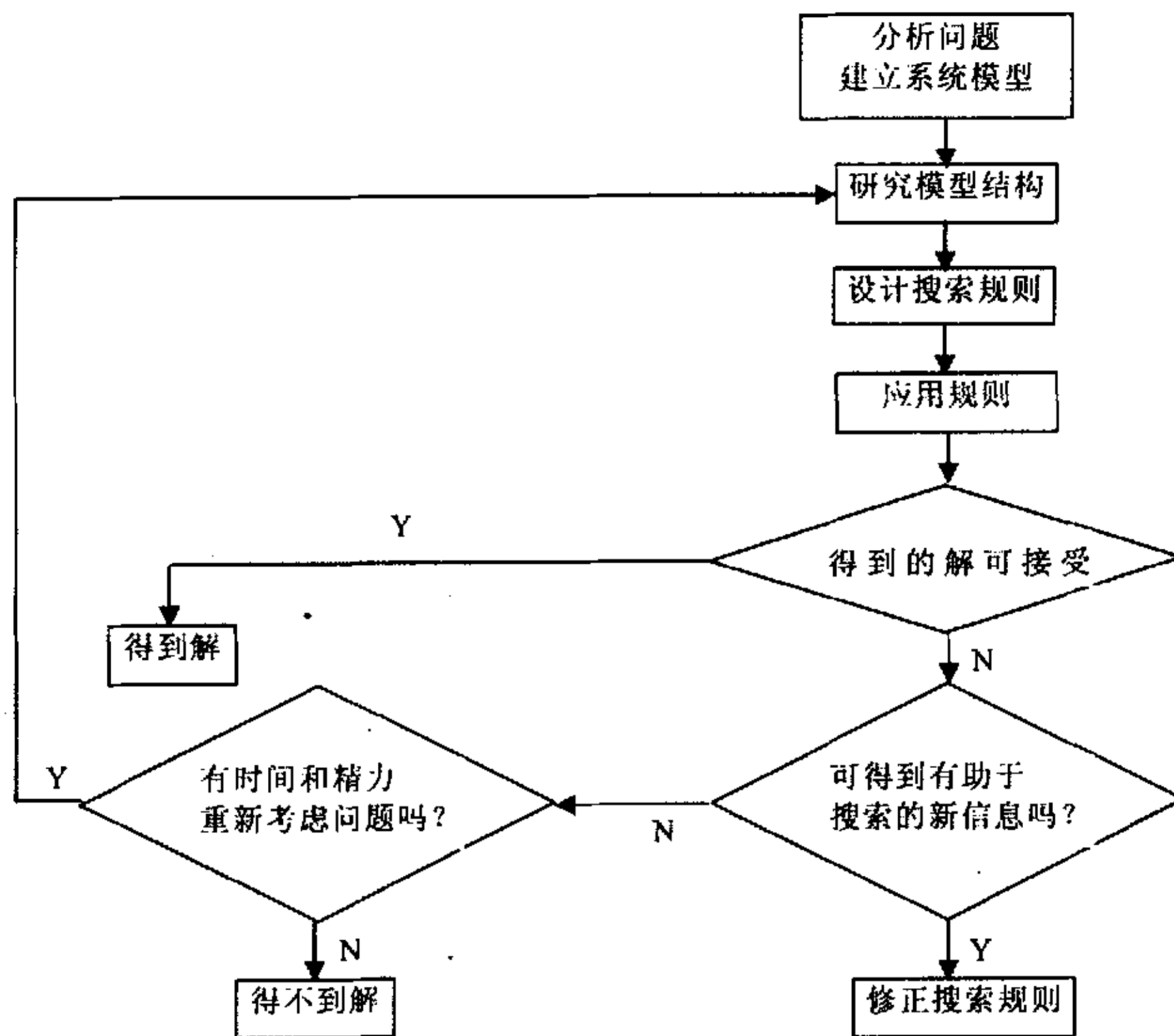


图 2-2 启发式方法求解流程

## 3. 启发式策略

用启发式方法求解问题时，需采用一定的策略。下面列出几个常用的策略，可根据问题的性质和要求选用。

### • 逐步构造解策略

逐步构造解策略是指每次增加解的一个元素(如节点、弧等)，直到得到一个完整的可接受的解。

### • 改进解策略

从一初始解(初始解不一定可行)开始，通过一系统替换分解和合并过程



来逐渐修正解, 以提高解的可接受性。

- 数学规划策略

运用数学模型和优化算法, 并通过对解的判别和修正以提高对问题的适用性, 学会得出高效的启发式算法。

- 分解策略

把一个复杂的问题分解成一系列易于处理的子问题来求解, 一子问题的输出常是下一个子问题的输入。

- 分割策略

把一个复杂的问题分割成一些平行的小的子问题, 然后求解每个子问题, 最后在相容原则下进行综合, 把子问题的解合并成原问题的一个子解。

- 可行解空间的限制策略

在某些情况下, 把可行解集限制在一个可应用已存在高效算法的解集上, 然后再求解问题。

- 松弛策略

有时扩展问题的可行域(即放松约束)以得到一个易于处理的松弛问题, 然后求解松弛问题, 就能直接得到或者很容易得到原问题的一个可行解, 然后再对得到的解进行修正。

- 搜索学习策略

包括在解空间的定向搜索以及在搜索过程中发现和收信新的信息, 并根据对新信息的分析, 重新确认或改变搜索方向, 修正搜索参数, 消去不必要的搜索范围, 以有效提高搜索效率, 尽快获得问题的解。

在构造启发式算法时, 一般都采用上述的某一策略, 或者是几种策略的组合形式。有时在启发式算法的某一步也可采用优化算法来提高算法的精度。

#### 4. 一种启发式方法求解过程介绍

对前面所提数学模型中式 2-1 应用启发式算法求解过程如下<sup>[5]</sup>。

(1)产生一个初始解本例算法基于贪婪算法形成。令顾客  $i, j$  之间的距离是  $c_{ij}$ , 对于一个给定的车辆  $k$ , 选择一个离中心车库最远的顾客  $i_1$  和一条最短的弧

$(i_1, i_2)$ , 满足  $x_{i_1 i_2 k} = 1$  和  $q_{i_1} + q_{i_2} \leq Q_k$ 。在这个算法的任一点, 我们将有一个

分枝路线  $i_1, \dots, i_m$ , 对车辆  $k$  来说满足  $x_{i_1 i_2 k} = x_{i_2 i_3 k} = \dots = x_{i_{m-1} i_m k} = 1$ 。在这个点

上, 我们选择一个最短的边连接点  $i_1$  或点  $i_m$  满足  $x_{i_1 j k} = 1$  或  $x_{i_m j k} = 1$  并有

$q(\{i_1, \dots, i_m, j\}) \leq Q_k$ 。当找不到可以满足所需的条件的边时, 再从未被访问的点(顾客)选择一条路径, 同样选择最远的点重复以上的过程, 产生一个分枝车辆路线, 直到所有的顾客都被访问为止。令  $f(y_k)$  表示在顾客集  $N(y_k) = \{i | y_{ik} = 1\}$  的旅行售货员问题的费用, VRP 的初始解为  $\sum_k [f(y_k) + M_k]$ 。

(2) 细化(Refining Procedure)通常  $M_k$  比较大, 为了减小总费用通常用增加车辆的做法是不可取的, 所以要在使车辆的数目最小的情形下优化车辆路径问题。不同路线的顾客交换和同一路线的费用优化可以达到减小车辆和总费用的目的。

(a) 不同路线的顾客的交换通常用下面方式进行, 如图 2-3。

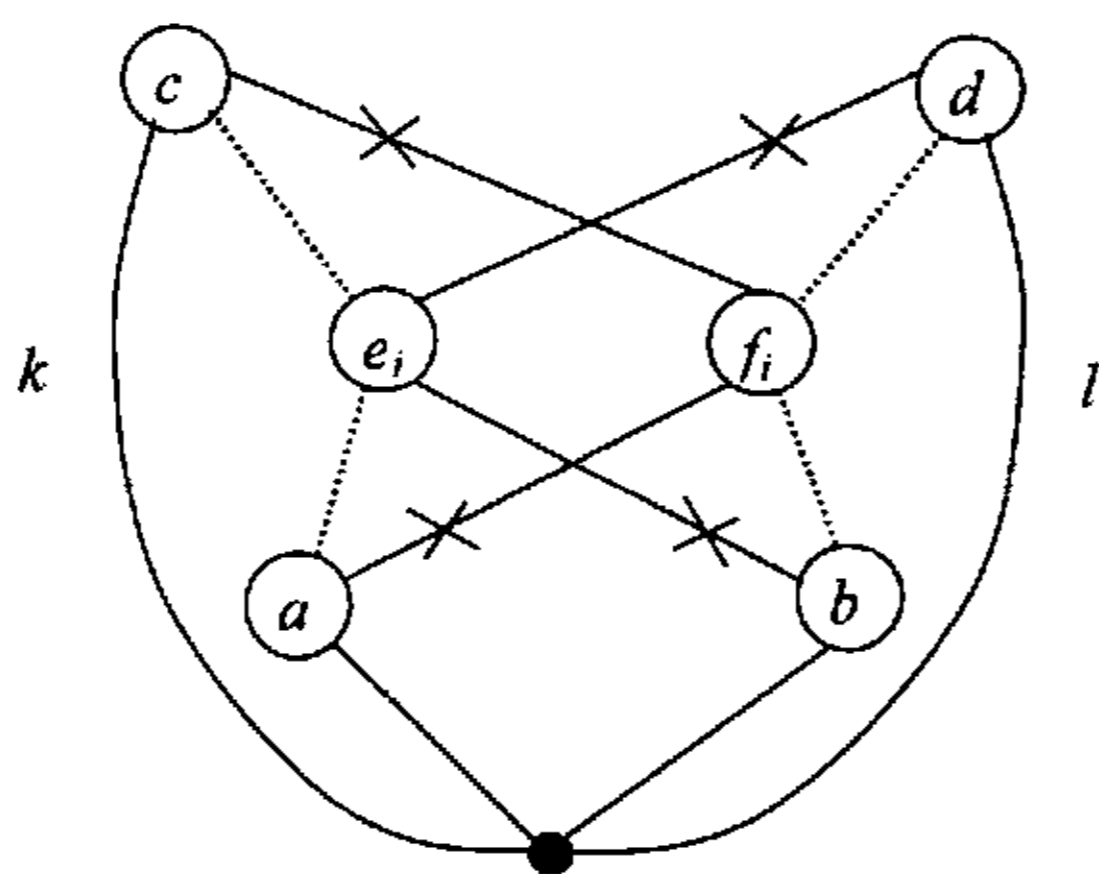


图 2-3 顾客交换过程

为了描述方便起见, 拟给初始解重新编号使  $\sum q_i y_{ik} - \sum q_i y_{i(k+1)} \geq 0 \quad k=1, \dots$ 。

设  $a$  是  $\sigma(f_i)$ ,  $\sigma(f_i)$  为第  $k$  条路线  $f_i$  前面一点; 设  $b$  是  $\pi(f_i)$ ,  $\pi(f_i)$  为第  $k$  条路线  $f_i$  后面的一点;

设  $c$  是  $\sigma(e_i)$ ,  $\sigma(e_i)$  为第  $l$  条路线  $e_i$  前面一点; 设  $d$  是  $\pi(e_i)$ ,  $\pi(e_i)$  为第  $l$  条路线  $e_i$  后面的一点;

如图 2-3, 路线  $k$  和  $l$  ( $l \geq k+1$ ) 的两个点  $e_i$  和  $f_i$ 。

若  $q_{e_i} > q_{f_i}$  且  $\sum_i q_{i \in s_k} - q_{f_i} + q_{e_i} \leq Q$  ( $s_k$  为  $k$  路线的顾客集); 令

$Q_i = q_{e_i} - q_{f_i}$   $C_i = C_{ae_i} + C_{e_i c} + C_{bf_i} + C_{f_i d} - C_{af_i} - C_{f_i c} - C_{be_i} - C_{e_i d}$ ; 对路线  $k$  ( $k=1,$

2.....)若  $\sum_i Q'_i \geq Q_k$ ; 且  $\sum_i C'_i \leq M_k$  交换所有  $e_i$  和  $f_i$ , 否则不交换。这样就节省费用( $M - \sum_i C_i$ )。重复上面过程直到找不到可交换的顾客为止。

(b)上面顾客交换是在不同的回路(不同的车辆路线)之间进行的, 在同一个回路之间的优化, 可以用分枝定界法进行。事实上, 同一个回路路线费用的优化问题是一个旅行商问题(TSP)。

### 2.2.3.2 遗传算法

#### 1. 遗传算法的工作机理

遗传算法是属于现代启发式算法中的一种, 是从达尔文进化论中得到灵感和启迪, 借鉴自然选择和自然进化的原理, 模拟生物在自然界中的进化过程所形成的一种优化求解方法。尽管这种自适应寻优技术可用来处理复杂的线性、非线性问题, 但它的工作机理十分简单。

标准遗传算法(Canonical Genetic Algorithms)的步骤如下:

- (1) 构造满足约束条件的染色体。由于遗传算法不能直接处理空间中的解, 所以必须通过编码将解表示成适当的染色体。实际问题的染色体有多种编码方式, 染色体编码方式的选取应尽可能地符合问题约束, 否则将影响计算效率。
- (2) 随机产生初始群体。初始群体是搜索开始时的一组染色体, 其数量应适当选择。
- (3) 计算每个染色体的适应度。适应度是反映染色体优劣的唯一指标, 遗传算法就是要寻得适应度最大的染色体。
- (4) 使用复制、交叉和变异算子产生子群体。这三个算子是遗传算法的基本算子, 其中复制体现了优胜劣汰的自然规律, 交叉体现了有性繁殖的思想, 变异体现了进化过程中的基因突变。
- (5) 重复步骤(3)、(4)直到满足终止条件为止。

#### 2. 遗传算法的特点

同传统的寻优算法比较, 遗传算法具有以下特点:

- (1) 遗传算法对问题参数的代码集起作用, 而不是对参数本身起作用。遗传算法处理的对象是染色体, 因而要求把所要优化问题的基本参数转化成定长的有限符号的染色体。
  - (2) 遗传算法是从初始群体搜索的, 而不是从单点开始搜索的。许多传统优
-

化方法都是从搜索空间的单点出发, 通过某些转换规则确定下一点。这种点到点的搜索方法在多峰值优化问题中, 首先找到的可能不是最优峰值; 而遗传算法是以点集开始的寻优过程, 初始群体是随机地在搜索空间中选取的, 这样在搜索过程中达到最优峰值的概率远大于到点方法的概率。

(3) 遗传算法在搜索过程中只使用适应度的函数信息, 而不是导数及其他辅助信息。对于不同类型的优化问题, 传统方法需要不同形式的辅助信息, 没有一种优化方法能适应各类问题的要求。遗传算法在优化过程中, 放弃使用这些辅助信息, 具有广泛适应性。

(4) 遗传算法使用概率转换规则而不用确定性规则。遗传算法使用概率转换规则来调整其搜索方向, 各代群体间没有统一的联系规律。但使用概率转换规则并不意味着这种方法属于随机算法范畴, 它只是使用随机转换作为工具来调整搜索过程趋向于目标函数不断改进的区域。

与传统方法相比, 遗传算法的优越性主要表现在首先, 在遗传算法的作用下, 遗传算法具有很强的搜索能力, 能以很大概率找到问题的全局最优解; 其次, 由于它固有的并行性, 能有效地处理大规模的优化问题。

### 3. 遗传算法的应用过程

将遗传算法应用于实际问题中, 通常包括以下过程:

- (1) 寻找有效且通用的编码方法, 将问题的可能解直接或间接地编码成有限位字符串。
- (2) 产生一组问题的可行解。
- (3) 确定遗传算子的类型, 选择所使用的操作算子, 如选择(select)、交叉(cross)、重组(combination)、复制(copy)、漂移(drift)等等。
- (4) 对操作算子选择其符合实际优化问题的模式, 如选择的模式、交叉的模式、变异的模式等等, 然后加以设计。
- (5) 确定算子所涉及的参数, 如群组的大小  $n$ 、交叉概率  $p_c$ 、变异概率  $p_m$  等等。
- (6) 根据编码方法和问题的实际需求, 定义或设计一个适应度函数, 以测量和评价各个解的性能的优劣。
- (7) 确定算法的收敛判据。
- (8) 根据编码方法, 设计译码方法。

### 4. 一种基于改进遗传算法求解的介绍

- (1) 建模
-

假定配送中心最多可用  $k$  辆车对  $l$  个顾客进行配送, 每个车辆载重为  $b_k(k=1, 2, \dots, K)$ , 每个顾客的需求为  $d_i(i=1, 2, \dots, l)$ , 顾客  $i$  到顾客  $j$  的运距为  $c_{ij}$ , 设  $n_k$  为第  $k$  辆车所包含的顾客数(若  $n_k=0$  表示未启用第  $k$  辆车), 用集合  $R_k$  表示第  $k$  条路径, 其中的元素  $r_{ki}$  表示在路径  $k$  中的顺序为  $i$ (不包含配送中心)。令  $r_{k0}=r_{k(n_k+1)}=0$  表示配送中心, 则有如下表示的车辆路径问题的数学模型<sup>[6]</sup>

$$\min \sum_{k=1}^K \left( \sum_{i=1}^{n_k} c_{r_{k(i-1)}r_{ki}} \right) + c_{r_{kn_k}r_{k(n_k+1)}} \cdot \text{sign}(n_k - 1)$$

$$\text{s.t.} \quad \sum d_{r_{ki}} \leq b_k \quad k=1, 2, \dots, K \quad (2-14)$$

$$0 \leq n_k \leq l, \quad k=1, 2, \dots, K \quad (2-15)$$

$$\sum_{k=1}^K n_k = l \quad (2-16)$$

$$R_k = \{r_{ki} | r_{ki} \in \{1, 2, \dots, l\}, i=1, 2, \dots, n_k\} \quad (2-17)$$

$$R_{k_1} \cap R_{k_2} = \emptyset \quad \forall k_1 \neq k_2 \quad (2-18)$$

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

上式中不等式(2-14)保证每条路径上的各顾客的总需求量不超过此条路径的配送车容量, 不等式(2-15)表明每条路径服务的顾客数不超过总顾客数, 等式(2-16)要求每个顾客都得到车辆的配送服务, 等式(2-17)表示每条路径的顾客组成, 等式(2-18)则限制每个顾客的需求仅能由一个车辆来完成。

## (2) 利用遗传算法解决 VRP 问题过程

从上述模型可知, 求解车辆路径问题的关键是合理确定车辆与各顾客的关系, 在满足车辆载重量和顾客需求约束条件的情况下使得总费用最小, 因此可以构造遗传算法如下

### 步骤 1 构造染色体, 产生初始种群

用矢量  $(s_1, s_2, \dots, s_l)$  表示染色体  $G$ , 其中元素(基因)  $s_j$  为  $[1, k \times l]$  之间的一个互不重复的自然数, 它表示了第  $j$  个确定第  $m = (s_j - \left\lfloor \frac{s_j - 1}{l} \right\rfloor \cdot l)$  个顾客与



路径  $k=(s_j - \left\lfloor \frac{s_j - l}{l} \right\rfloor + 1)$  的关系( $\lfloor \cdot \rfloor$ 表示取整数, 下同。), 即确定顾客  $m$  是否

由车辆  $k$  配送及确定分库  $m$  在路径  $k$  中的顺序的次序为  $j$ 。随机产生一组染色体  $G_h(h=1, 2, \dots, n)$ (其中  $n$  为一代种群中的个体数),  $G_h$  各不相同, 此为第一代种群。

### 步骤 2 可行化过程

将染色体的编码向量映射为满足全部约束条件的可行解称为可行化, 其过程如下

(I) 令顾客需求条件满足的标志变量  $d_{zm}=0(m=1, 2, \dots, l)$ 。

(II) 令路径  $k$  中的顾客数目  $n_k=0(k=1, 2, \dots, K)$ , 令  $b'_k=b_k$ ,  $R_k=(k=1, 2, \dots, K)$ , 路径  $k$  中除去配送中心后第  $i$  个位置的顾客号为  $r_{ki}=0(i=1, 2, \dots, l)$ , 即此时所有路径皆未形成。

(III)  $j=1$ 。

(IV) 第  $j$  次确定顾客  $m$  与路径  $k$  间的关系, 其中,  $m=(s_j - \left\lfloor \frac{s_j - l}{l} \right\rfloor \cdot l)$ ,

$$k=(s_j - \left\lfloor \frac{s_j - l}{l} \right\rfloor + 1)。$$

(V) 判断  $d_{zm}$  是否等于 0, 若等于 0, 表明顾客库  $m$  的需求尚未满足, 转(V) 继续判断路径  $m$  的情况; 否则转(VII)。

(VI) 判断  $d_{zm}$  是否为 0?

① 若为 0 判断  $d_m \leq b'_k$  成立否? 若成立, 令  $d_{zm}=1$ ,

$b'_k = b'_k - d_m, n_k = n_k + 1, r_{km} = m, R_k = R_k \cup \{m\}$ ; 若不成立, 转(VII)。

② 若不为 0 转(VII)。

(VII)  $j=j+1$ , 转(IV)重复上述过程, 直到  $j=K \times l + 1$ 。此时检查是否所有  $d_{zm}=1$ , ( $m=1, 2, \dots, l$ )成立, 若成立, 说明在满足各约束条件的情况下, 所有的顾客均分配了一个路径, 构成路径集合  $RT=\{R_1, R_2, \dots, R_k\}$ , 即为染色体所对应的原车辆路径问题的一个可行解; 若不成立, 说明此染色体表示的路径分配方案不满足约束, 为原 VRP 问题的一个不可行解。

**步骤 3 性能估计**

对一代种群中的每一个染色体  $G_h (h=1, 2, \dots, l)$  应用步骤 2, 求得对应可行解  $RT_h (h=1, 2, \dots, n)$ , 代入目标函数

$$Z_h = \sum_{k=1}^K \left[ \sum_{i=1}^{n_k} (c_{r_k(i-1)r_k} + c_{r_k r_k(n_k+1)}) \cdot \text{sign}(nk-1) \right];$$

若染色体对应的

为非可行解, 则赋予其目标函数一个很大的整数  $z_h=M$ 。令  $G_h$  的适应性函数  $f_h=1/z_h$ ,  $f_h$  是个体  $G_h$  在生存竞争中生存能力的表现, 越大表明其性能越好, 即其对应的解越接近最优解。

**步骤 4 判断停止进化条件**

判断迭代的代数是否为要求代数  $N$ , 若是, 停止进化, 选性能最好的染色体  $G_h^*$  所对应的路径集合  $RT_h^*$  作为原 VRP 问题的优化解输出。反之, 继续执行步骤 5。

**步骤 5 自然选择**

将每代种群共  $L$  个染色体按适应值  $f_h$  由大到小排列 ( $h=1, 2, \dots, n$ ), 排在最前一位的个体性能最优, 将它复制一个, 直接进入下一代种群。下一代种群的另  $L-1$  个染色体则从前代种群的  $n$  个染色体中按概率  $p_h=q'(1-q)^{h-1}$  ( $h=1, 2, \dots, n$ ), 用轮转法选择个体  $G_h$ , 产生后代形成。这样既可保证最优者生存至下一代, 又可避免个体间因适应值大小不同而使被选择进入下一代的机会相差悬殊, 保持了下一代种群个体的多样性, 从而可有效提高整个算法的收敛速度。其中  $q'=q/(1-(1-q)^n)$ 。

**步骤 6 染色体交叉重组**

对步骤 5 所门生的新种群, 按选择概率  $p_c$  选择个体对进行交叉重组, 共进行  $n/2$  次。相关文献[11]中说明  $p_c=0.6\sim 0.8$  之间时, 进化性能较好。

**步骤 7 染色体变异**

在每代种群中, 以变异率  $p_m=C$  ( $C$  为已定的一概率值) 对染色体进行变异, 变异策略是随机交换选中染色体两个基因的值, 对变异成功所获染色体应用步骤 2 与步骤 3 求其适应值, 并与其父染色体比较, 择性能优者进入种群。

**步骤 8 返回步骤 4, 循环。**

**2.3 带时间窗的 VRP 问题(VRPTW)****2.3.1 时间窗定义**

VRPTW 问题是在传统的车辆路径问题上加上时间窗的约束, 时间窗约束可分为硬时间窗、软时间窗和混合时间窗三种<sup>[7]</sup>。

(1)硬时间窗(Hard Time Windows)指配送车辆必须在规定时间段内(如图 2-4 中 $(t_1, t_2)$ 段)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。当配送货物送达时间超过了规定的时间段 $(t_1, t_2)$ , 规定一个惩罚值 $P(t)$ ( $P(t)>0$ )非常大的惩罚函数(Penalty Function), 以表示硬时间窗的限制。

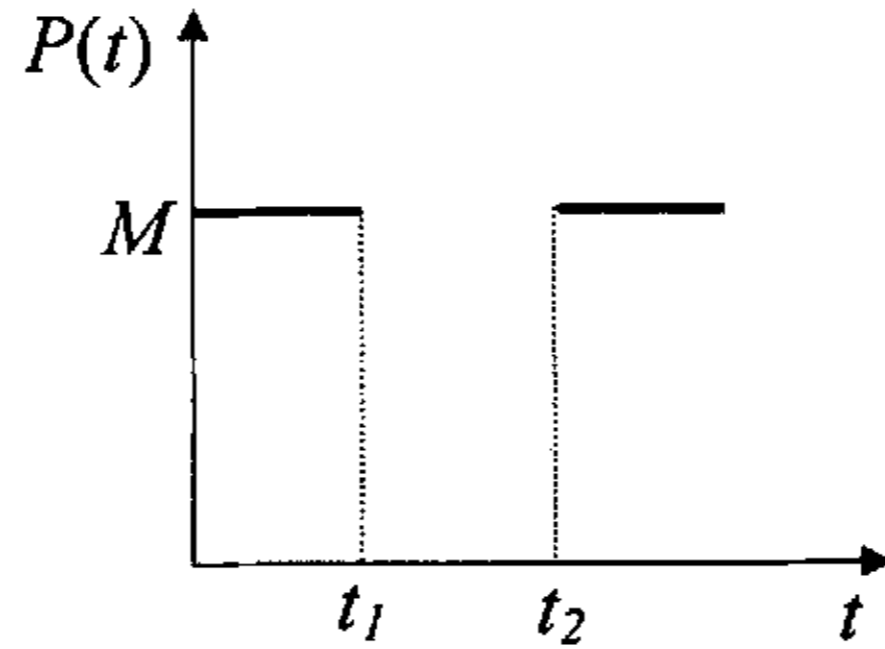


图 2-4 硬时间窗

(2)软时间窗(Soft Time Windows)指配送车辆如果无法将配送货物在特定的时间段(如图 2-5 中 $(t_1, t_2)$ 段)内送达到顾客手中, 则必须按照违反时间的长短施以一定的罚金或其它处罚来补偿由之而为客户带来的损失的法则。图 2-5 就是一种可能的处罚函数。

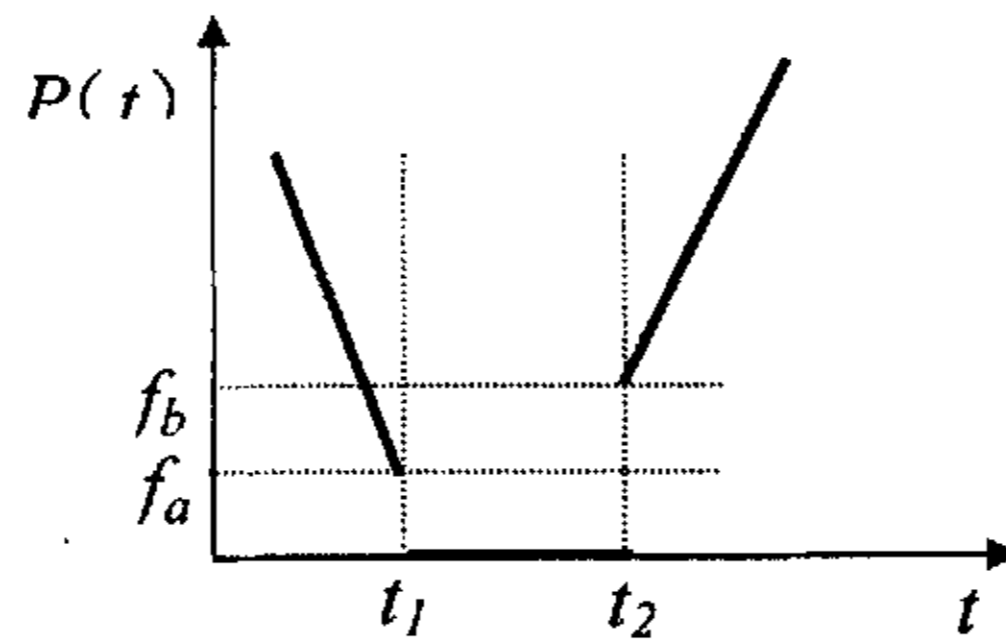


图 2-5 软时间窗

(3)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。在实际的配送工作中, 本着车辆如果能在最佳时间段(如图 2-6 中 $(t_1, t_2)$ 段)将配送货物送达到顾客手中, 则不处罚; 若在 $[(t_0, t_2)$ 或 $(t_2, t_3)]$ 时间段内送达, 则顾客的满意度下降, 处以一定的处罚(转化为处罚函数), 并且顾客不接受在时间段 $[(t_0, t_1)$ 或 $(t_3, \infty)]$ 中的任何时间内的收货。

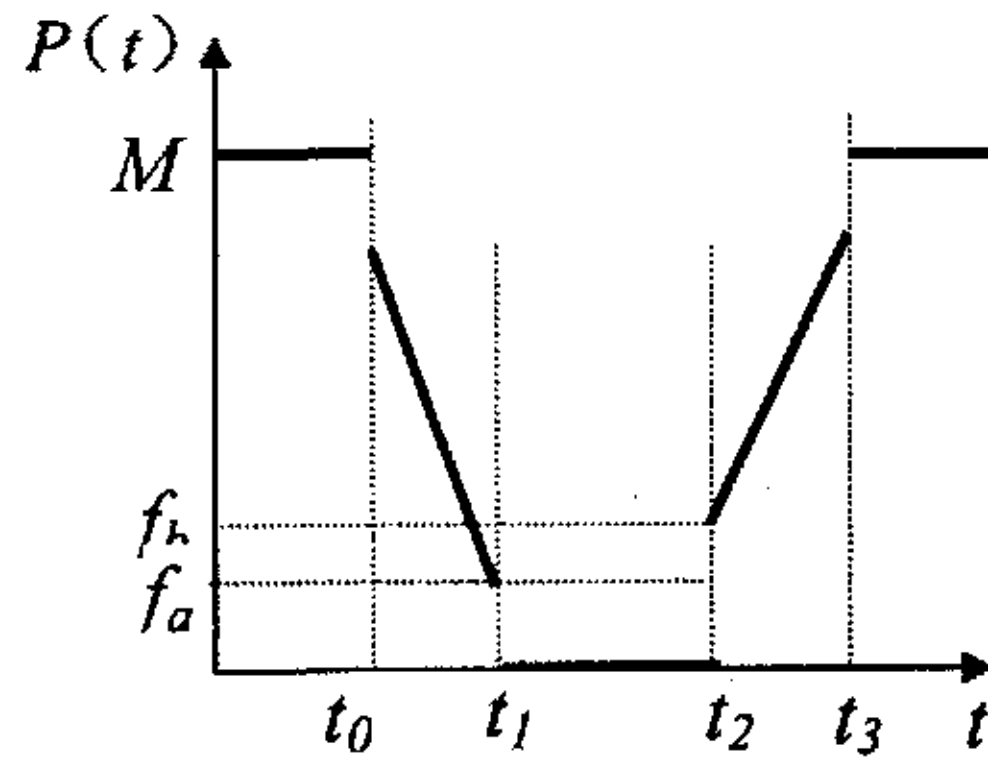


图 2-6 混合时间窗

### 2.3.2 求解 VRPTW 问题的关键因素

含时间窗的 VRP 问题可以看作是指派问题(Assignment problem; AP)的拓展。由于时间窗约束的车辆调度问题是在传统的 VRP 问题的基础上再加入顾客的时间窗约束, 所以需要对以下三项关键问题进行决策

#### (1) 路径查找(Routing research)

VRP 问题中的配送车辆都由配送中心出发, 而配送中心在满足顾客需求的基础上以追求最小成本为目标, 所以配送中心必须为每辆配送车指定配送路线及顾客, 然后回到配送中心。这一指定的线路对每一车辆来说, 是一个旅行商问题(TSP);

#### (2) 装载量限制>Loading confine)

对于配送中的每一配送车辆都有规定负荷的载容量限制, 所以在配送方案制定过程中应该对车辆载容量加以限制, 这时考虑车辆配送路径才是有意义的;

#### (3) 顾客服务次序(Customer service order)

含有时间窗的 VRP 问题, 必须考虑各顾客对服务时间的要求, 所以对顾客的服务次序的确定也是 VRPTW 所要决策的问题。

### 2.3.3 含时间窗的 VRP 问题算法分析

通过对近年来主要研究用于解决 VRPTW 问题文献的总结, 对 VRPTW 问题的常用解法有四种。

#### 1. 精确优化方法(Exact procedures)

精确优化法将配送问题, 通过严谨的数学模型或计算机数据结构规划,

利用数学法则或数据结构搜寻的方式,求得问题的解。由于两种方法都在所有可行解集合(Feasible Solution Set)寻找最优解,所以所求的解均为最优解。随着变量的增加,配送问题的解集合会有组合爆炸(Combinatorial Explosion)的现象,求解时间也呈指数函数的趋势增长,不能在有限的时间(Polynomial Time)给予决策者一个可行解,是这种方法最大的缺点。常见的精确解法有动态规划法(Dynamic programming)、分枝定界法(Branch and bound)与切平面法(Cutting planes)。

#### (1)动态规划法(Dynamic programming)

动态规划法是美国学者贝尔曼(Bell man)在 20 世纪 50 年代提出的一种动态最优化方法。它的基本原理是,一个多级决策的最优策略具有这样的性质,即无论其初始状态和初始决策如何,从这一决策后所导致的新状态开始,以后的一系列决策必须是最优的<sup>[9]</sup>。这就是构成动态规划模型的条件之一状态变量必须满足“无后效性”,即如果其状态给定,则在这段以后过程的发展不受前面各状态的影响,亦即过程的过去历史只能通过当前的状态去影响它未来的发展,当前的状态就是未来过程的初始状态。此方法的核心观念为最佳化原理,其运用数学逻辑来处理一连串相互关联的决策问题,并采取系统化步骤以求得对整体最有利的策略。求解方法与步骤如下所述

**步骤 1** 将整个问题分解为许多相互关联的局部问题,以便个别分析处理。

**步骤 2** 从最后一个阶段的各种状态中,找出最有利的决策方案。

**步骤 3** 利用“反向推算法”,自最后一个阶段逐步向前项阶段推进,直到各阶段的最有利策略均找出为止。

Held 和 Karp<sup>[12]</sup>运用此法于求解巡回推销员问题,因其计算时间与计算机内存空间均随变量的增加而呈指数增加,所以虽然此方法可求得最优解,但仅适用于规模较小的问题。

#### (2)分枝定界法(Branch and bound)

分枝定界算法(Branch-Bound Algorithm),简称 B&B 由 Land Doig 等人于本世纪六十年代提出<sup>[13]</sup>。其算法思想适用于表达成整数线性规划(或者混合整数线性规划)的问题。它采用了类似分而治之的算法策略,在分析问题的一切可行解的过程中,采取了必要的限制条件,设法排除可行域中大量非最优解区域,从而能够有效求解一些规模较大的问题。

**算法步骤** 算法过程中,将所有待分解或探查的节点(子问题)存储于集合 active set 中, U 中存放目前发现的最优解值。



步骤 1 初始化(initializing step)。U= $\infty$ 。尽可能去掉一些明显的非最优点，将其余的可行解集做为一个子集合。转到步骤 2。

步骤 2 分枝(branching step)。采用某种分枝规则，从目前的若干可行解子集中选择一个子集，将其分解为若干子集合。

步骤 3 定界(bounding step)。对每个新子集，计算其对应的下限。

步骤 4 探查(fathoming step)。根据计算所得到的下限值的估计值进行判断，决定是否进一步分解子集，情况如下①下限值大于等于目前最优解值；②新子集中不含可行解，从可行域中排除这个新子集；③对问题求到最优解，此时新的对应的下限值产生。如果①成立，那么从可行域中去掉新子集；如果①不成立，则令目前最优解等于新子集所对应的下限值，将得到的解作为当前最好解，然后，对其它节点(子问题)再根据①进行如上判断。

步骤 5 停止规则(stopping step)如果没有节点(子问题)待处理，即 active set 不含点集，算法终止，当前所获得的最好解为最优解。否则返回到步骤 1。

### (3)切平面法(Cutting planes)

此方法与分枝界限法类似，也是在求解与整数规划相对应的线性规划上，不断地增加新的约束，也就是另外加入线性约束条件，以切掉对应于非整数规划的所有可行解的集合，以使问题可达到整数线性规划求解的形式，从而获得最优解。

上述三种精确解法的优点在于可求得最优解，缺点在于求解时间过长。

## 2. 启发式算法(Heuristics)

启发式算法是通过经验法则来求取运输过程满意解的数学方法。启发式方法在解题时可减少搜寻的次数，能同时满足详细描绘问题和求解的需要，较精确优化方法更为简单实，是一种容易且快速求解困难问题的算法。Solomon<sup>[14]</sup>曾提出几个时间窗约束车辆巡回问题的启发式解法，包括节省法(Saving method)、最邻近法(Nearest neighbor)、插入法(Insertion)及扫描法(Sweeping)。由于上述各解法起初都用于求解传统 VRP 问题，如果应用于求解 VRPTW 问题时，则在求解过程中必须增加时间可行性的检验，此检验的目的在于确保配送行程能满足顾客的时间窗的要求；所以当顾客被排入路径时，需检验该排入时点是否违反该顾客与其后受影响的顾客的时间窗的限制。以下针对各启发式解法进行介绍。

### (1)节省法

Clarke & Wright<sup>[15]</sup>于 1964 年提出该方法以求解车辆路径问题，其思想

在于按节省值(较短路径与原路径之差)由大至小排序,在车辆容量限制下,依序将对应的两顾客点排入路径中,直至所有顾客都被排入路径为止。Solomon 于 1983 年将此法应用于求解时间窗约束的车辆巡回问题,关键在于当节省值较大的两顾客点被排入路径时,除需考虑车辆容量限制外,更需要考虑到时间窗的限制,也就是时间窗上界较早者,应优先被配送,并检验其时间可行性,此方法的优点是可提高车辆的利用率。而两节点间的节省值的计算公式与意义如下所示

$$s(i, j) = d(i, 0) + d(0, j) - d(i, j) \quad (2-14)$$

其中  $d(i, 0)$  代表顾客  $i$  至配送中心的距离,  $d(i, j)$  则代表顾客  $i$  至顾客  $j$  的距离。计算两节点  $i$  与  $j$  间的节省值  $s(i, j)$  时,应先计算原路径中各往返路径的总和,再以之与较短路径的总路径和相比较;两节点的原路径与较短路径如图 2-7 所示。

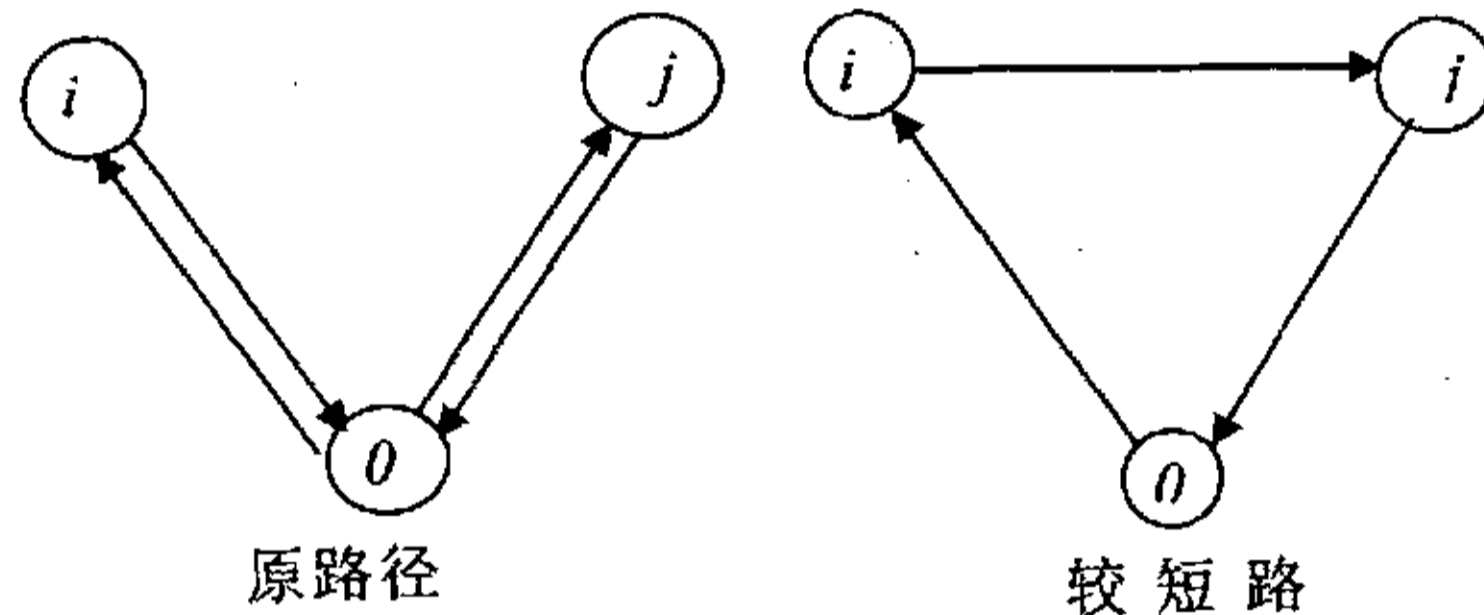


图 2-7 节省法图示

## (2) 最邻近法

Solomon 于 1983 年所提出的求解 VRPTW 的方法,该方法的思想如下所述:

由配送中心开始搜寻,若  $i$  为离配送中心最邻近的节点( $c_{ij}$  最小),则其优先排入路径中,接着搜寻另一节  $j$  以排入路径的下一位,且该节点加入后必须满足以下条件,满足者称为可排入点

- ① 之前尚未被排入路径的节点
- ②  $c_{ij}$  最小者
- ③ 车辆载量限制与时窗限制的时间可行性

若其它尚未排入路径的节点中皆无法找到可排入点,则构造另一新路径,直到所有顾客点都被排入所有路径为止。而决定节点是否为最邻近点的要素有以下三点

- ① 两顾客节点间的距离  $d_{ij}$

- ② 两顾客节点间的运输时间  $t_{ij}$
- ③ 配送下一顾客时所需等待服务的时间  $u_j$

在决定两节点间的最邻近法总成本时, 配送中心应先评估对此三要素的权重组合, 且总和应为 1, 所以最邻近法总成本的计算公式如下所示

$$c_{ij} = \delta_1 d_{ij} + \delta_2 t_{ij} + \delta_3 u_j$$

其中  $\delta_1 + \delta_2 + \delta_3 = 1$  且  $\delta_1$ 、 $\delta_2$ 、 $\delta_3$  为配送中心对该三要素的权重组合。

### (3) 插入法

插入法原本是 Mole 和 Jameson<sup>[16]</sup>于 1976 年所提出, 用于求解 VRP 问题的方法, 其结合最邻近法与节省法的理论, 依次将顾客点插入到路径中以构建配送路线。Solomon 于 1983 年将此方法应用于求解 VRPTW 问题, 因为时间因素加入, 而使原问题的顾客等待时间缩短。此方法包含二个步骤

步骤 1 选取距离配送中心最远的顾客点为起点, 从其它剩余的顾客点中, 根据最邻近法决定下一个被插入的顾客点。

步骤 2 以节省法决定该顾客点应被插入的位置, 在车辆容量限制下, 重复进行选取与插入的步骤, 当无法再扩充路径时, 则再建立另一路线, 直至所有顾客都被排入路径中。

### (4) 扫描法

Gillett 和 Miller<sup>[17]</sup>于 1974 年所提出的求解 VRP 问题的方法, 此方法属于先分群再排路线的方式, 此方法分为两阶段

第一阶段利用极坐标来表示各需求点的区位, 然后任取一需求点为起点, 以车辆容量为分群的约束, 再以该需求点为零度按顺时针或逆时针的方向, 进行顾客的扫描分群。

第二阶段依据求解旅行商问题的算法, 求解各顾客群的路线安排。

Solomon 于 1983 年将此法应用于求解 VRPTW 问题, 与原扫描法不同点在于第二阶段的求解各顾客群路线安排, 其以插入法进行各顾客群的路线安排, 并检查时间可行性, 若有顾客点无法满足时间窗的约束, 则先排除此顾客点。若所有的顾客群都已安排, 且所有的顾客点都被服务, 则完成路线的建构; 若有顾客点尚未被服务, 则沿原扫描方向, 将剩余的尚未服务的顾客点重复进行扫描与插入的步骤, 直到所有的顾客点都被服务。

## 3. 模拟方法(Simulation)

### 4. 交互式优化法

---

## 2.4 考虑库存量的 VRP 问题

### 2.4.1 问题分析

传统的配送方案的制订主要是依据顾客订单,即配送产品种类、数量及配送时间是由顾客确定并通过订单下达给配送中心的,配送中心在配送方案的制订过程中,只考虑配送成本。而随着供应链一体化思想的不断发展与成熟,在现代的配送中心制订配送方案过程中,管理者需要在保证顾客服务水平的前提下,根据对顾客需求的预测情况确定配送产品种类、数量、配送频率、送货时间、配送方案,而非由顾客通过订单下达指令,管理不仅要考虑如何降低配送时产生的配送成本,还要考虑顾客的库存成本,即要尽量满足顾客的货物需要,尽可能少的产生缺货所带来的损失,又要尽量减少顾客的库存费用,这就是近年来被广泛研究的考虑顾客库存量的 VRP 问题。

### 2.4.2 主要研究方法分析

在制订配送方案的时候把顾客的库存量同时考虑进去最早在七十年代被提出过,如 Herron<sup>[23]</sup>在 1979 年曾论述同时优化库存和运输问题的重要性和由之所带来的经济效益,但是并没有提出具体解决办法。把库存与配送整合为一个问题的研究是从 80 年代开始的,如 Federgruen 和 Zipkin 于 1984 年<sup>[24]</sup>首次用一个模型同时表述配送与库存这两个问题。

由于配送与库存是运筹学中两个重要的课题,近些年来,关于研究它们各自的文献及专著都十分丰富,但是把两者结合起来研究的资料相对较少,近年来对这一问题的研究主要包含在运输—库存联合优化研究(IRP)中。运输—库存联合优化问题从系统化、集成化的思想出发,将物流的两大功能,运输调度与库存控制作为一个整体进行优化,不仅考虑运输成本,还要考虑各项库存成本,确定配送对象(或进货来源)、运输产品种类、数量及运输车辆分配和路线优化方案,以实现整个系统的成本最小化。

Federgruen 和 Zipkin 对考虑顾客库存的 VRP 问题的研究是从车辆调度问题(VRP)的优化方法中发展而来的,解决随机型需求模式下单—供应商的考虑库存的 VRP 问题。在建立的非线性整数规划模型中,目标函数是当日的总成本最小,包括运输成本、库存成本及全天缺货成本;约束条件包括供应商库存能力限制、客户库存能力限制及变量非负限制等。由于受到库存能力限



制并考虑库存成本与缺货成本,并非每天向所有客户送货,为了简化问题,引入一条虚拟路径,所有得不到送货的客户均包括在此路径内。非线性整数规划能够将配送优化问题分解为库存分配问题和旅行商问题(TSP),通过前者优化库存成本和缺货成本,后者则优化运输成本。其解法的主要思路是首先确定一个初始可行解,然后通过相互交换不同配送路线中的客户改变条件,反复迭代,不断优化结果。显然,这种方法要比标准的 VRP 问题的计算量大得多。这主要体现于不同配送路线中客户的交换中,每交换一次也就定义了一个新的库存分配问题和新的 VRP。

Golden, Assad 和 Dahl 采用启发式方法(Heuristics)解决这一问题,即在充分满足所有客户存货需求的前提下使总成本最小。在这种方法中,定义了“需求迫切性”,即库存能力与实际库存水平之比。首先,分别计算每个客户的需求迫切性,并将之与事先确定的标准(最小值)进行比较,直接排除该比值低于标准的客户。其次,根据需求迫切性与所需送货时间的最高比率,选择客户直接配送。然后,通过反复迭代建立一个大型 TSP。其中,全程时间(TMAX)的初始值不得超过配送车辆数量与日工作时间的乘积。在耗时达到此限制或者所有客户需求均已满足之后,不能再向其他客户送货。在不允许送货不足的情况下(即只要送货,必须完全满足收货客户的需求),得到一组可行路径,共同构成最终的优化配送线路:若此条件无法满足,则减少 TMAX 的值,再使用上述启发式方法。

Balakrishnan 和 Wong 提出一种不同于上述解法的改进方法。其不同之处体现于不将每一天视为独立的一个时间周期,通过信息传递,考虑前一天的优化结果对次日的影响,对于连续几天库存/配送情况进行模拟。假设每个客户的日最大用量已知,则通过送货单位收益和缺货单位损失可以计算日利润,这种方法也正是以日利润最大化为目标的。这种方法将问题描述为整数规划模型,解决如何向客户分配供应商的有限库存、车辆调度及路线选择问题,并运用拉格朗日对偶探测法(Lagrangeandual ascent method)解该整数规划。



### 第 3 章 数学模型的建立与求解

配送方案优化过程是一个相当复杂的过程,其中所涉及的相关问题和因素非常多。从研究对象来讲,配送方案可是供货方(可为工厂、仓库或配送中心等)和需求方(可为零售商、其它工厂或其它配送中心等)组成的一个系统中对车辆路线的安排与库存量的控制。分三种情况即 R-System(Retailer System), DR-System(Depot and Retailer System)和 D-System(Depot System)。这三个系统所涉及的运输优化方面基本相同,不同的是库存方面;从拓扑结构来讲,在配送方案优化过程中由供、需方所形成的系统的网络结构,可能是单——单(One-One)、单——多(One-Many)、多——单(Many-One)或多——多(Many-Many)。其中前者指供货方的个数,后者指需求方的数量,如单——单系指由一个供货方和一个需方构成的系统;从货物种类来讲,在供、需房间流动的货物种类的数量,可为一种货物或多种货物。当处理的是多货物种类的时候,一般也是先对每种货物单独处理;从费用因素来讲,主要涉及运输和库存相关费用。运输相关费用包括启用运载工具的固定费用,与走行距离相关的可变费用,有时也包括运载工具在需方停留的费用。库存相关费用主要由货物保管费、缺货损失费和订货费构成。从限制因素方面讲,涉及的具体问题不同,求解问题的限制也不同,但一般都包含几个主要因素,如运载工具的载重量,供方的供应能力,需方的存储能力,可执行运输任务的运载工具数量,需方的时间窗限制等。

#### 3.1 不考虑配送中心库存单周期配送方案优化

本节所研究的问题是在不考虑配送中心的库存成本的情况下,假设配送的货物为单货物品种、One-Many 的 R-System,客户的需求是随机的。设独立均匀分布于一个不规则区域中的由  $n$  个顾客和一个配送中心组成的单物品配送系统,彼此之间的距离  $d_{ij}(ij=0,1,2,\dots,n)$  已知。首先研究配送中心对一个顾客直接配送的情况。即一对一配送,再研究一配送中心对多个顾客配送的情况,即一对多配送。

##### 3.1.1 一对一配送方式分析

一对一的配送方式就是从一个配送点分别向每一个顾客配送货物的方式。这一部分首先对每单位货物的运输和库存成本提出了表述,然后评价了

---

在成本间作为运输量的函数的平衡。

### 1. 建模与分析

在一对一的配送方式选择下, 考虑单一顾客, 设

$q$ =顾客需求量	(货物量/天)
$P$ =平均单位货物价值	(元/单位)
$R$ =库存资产持有量	(每周货物价值的部分)
$D$ =配送点到顾客的运输距离	(公里)
$T$ =配送点到顾客的旅行时间	(单位时间)
$W$ =卡车运量	(货物单位数)
$\gamma$ =初始分配的固定成本	(元/车)
$\alpha$ =每单位距离的运输成本	(元/公里)
$\sigma$ =每次在顾客处停车一次的固定成本	(元/停)
$V$ =运输量	(货物单位数/车)

直接配送包括每车只停一次, 因此每车运输成本  $F$  为

$$F = \gamma + \sigma + \alpha D$$

单位货物的运输成本  $C_s$  为  $C_s = F/V$

库存成本依赖于对货物的生产和消费之间的时间, 如果货物独立于顾客计划生产(即生产计划与运输计划不一致), 并且顾客需要一种货物, 则需要产生一次运输量  $V$  的平均时间是  $V/q$ , 每单位货物在补充运输之前在车上的等待时间是  $V/q$  这一平均时间的一半(即  $V/2q$ )。

如果顾客以一个稳定的频率使用货物, 则, 用一车运输量  $V$  单位货物所需的平均时间也是  $V/q$ 。因此, 货物运到后, 每单位货物在用掉之前的平均等待时间也为  $V/2q$ , 则一单位货物从生产到消费, 平均用的总时间  $\tau$  为

$$\tau = V/q + T \quad (3-1)$$

每单位货物平均库存成本  $C_i$  为

$$C_i = PR(V/q + \tau) \quad (3-2)$$

与  $V$  相关的运输库存联合成本现在可以表述为

$$C = C_s + C_i = F/V + PR(V/q + T)$$

最佳运输量( $V^*$ )由下面得

$$V^* = \min(S, W) \quad (3-3)$$

这里  $S = \sqrt{Fq/PR}$   $W =$  卡车载容量

公式(3—3)说明最优的运输量可能比车辆满载的时候少, 当为这种情况的时候,  $V^*$  对于一个时期  $1/q$  (即  $PR/q$ ) 在仓库保存一单位货物的成本, 会随着固定运输成本  $F$  的比率而增加, 因此,  $V^*$  会随从配送点到顾客的距离而增加。与  $V^*$  相关的最小成本为

$$C^* = \begin{cases} 2\sqrt{PRF/q} + PRT & \text{如果 } S \leq W \\ F/W + PRW/q + PRT & \text{如果 } S > W \end{cases} \quad (3-4)$$

$C^*$  随着在配送点与顾客之间的货物流量  $q$  而减少, 同时, 也注意到, 每单位时间的总成本表现出,  $qC^*$  是经济规模, 除非  $q$  足够大, 以至于能说明运输满载车是合理的 (即  $q=W^2PR/F$ )。这一重要的对  $q$  关于  $qC^*$  的凹面的相关性, 在把线性规划用于对自然分配网络的流动配送建模的时候, 一般可忽略, 特别的是, 线性规划只有在关联的成本对流量为线性或分阶段线性凸时, 才能最优化配送。在现实中, 这种线性假设只有在  $V^*=W$  的时候才成立。当  $V^*<W$  时, 线性规划找不到真正的最优解。

由(3-4)在以上相关参数确定的情况下, 如果顾客需求量确定了的同时, 影响  $C^*$  的变量就是函数  $F$  中的配送点到顾客的运输距离  $D$ 。

## 2. 模型算法设计

通过以上分析, 原问题转化为一单源的最短路径的求解问题。对于这类问题, 目前应用最广泛的是 DIJKSTRA(1959)年提出的算法, 是当前普遍采用的一个较好的算法, 这个算法的主要思路是用逐点增长的方法构造一棵路径树, 从而得到从树要(指定点)到其他点的距离。下面先介绍这一思路。

设: 求从指定点  $v_0$  到某一点  $v$  的最短路径。

从  $v_0$  出发沿着弧的方向第一个到达而且路径最短的一点, 一定是所有与  $v_0$  有弧关联且弧的长度最小的那个点, 这是与  $v_0$  距离最短的一点, 称为第一短路径点, 如果这个点就是  $v$  则问题解决。如果不是  $v$  而是另一点  $v_1$ , 那么从  $v_0$  出发的第二条最短路径会到达哪一个点呢? 一定是与  $v_0$  有弧相关且弧的长度最小的那个点 ( $v_1$  除外), 或者是由  $v_0$  通过  $v_1$  再经过与  $v_1$  关联的长度最小的弧所到达的点, 设为  $v_2$ , 如果  $v_2$  即为所求的  $v$ , 问题即得到解决, 否则继续寻求从  $v_0$  出发第三短路径到达的点, 第四短路径到达的点, 如此继续下去直到路径的终点是  $v$  为止。

一般来说, 设  $T$  为以求得的第一、第二……第  $K$  条路径的终点的集合, 即  $T=\{v_1, v_2, \dots, v_k\}$ , 则有  $L(v_0, v_1) \leq L(v_0, v_2) \leq \dots \leq L(v_0, v_k)$ 。那么第  $K+1$  条短路

径(设其终点为  $w$ )或者是弧 $(v_0, w)$ , 或者终点是  $T$  外的某一点  $w$ , 但路径的中间结点一定都是  $T$  中的结点(即以到达过的结点)这是显然的, 如果从  $v_0$  到  $w$  的路径中有一中间点  $i$  不在  $T$  中, 由于  $L(v_0, i) < L(v_0, w)$  则  $i$  点必然作为  $v_0$  到  $w$  的路径更短的一条路径终点先行加入  $T$  中。因此按照这种路径逐步增长的过程, 我们将得到一棵以  $v_0$  为根的树, 如图 3-1 所示, 称为路径树, 从树根到各结点的路径长度即是  $v_0$  到该点的距离。

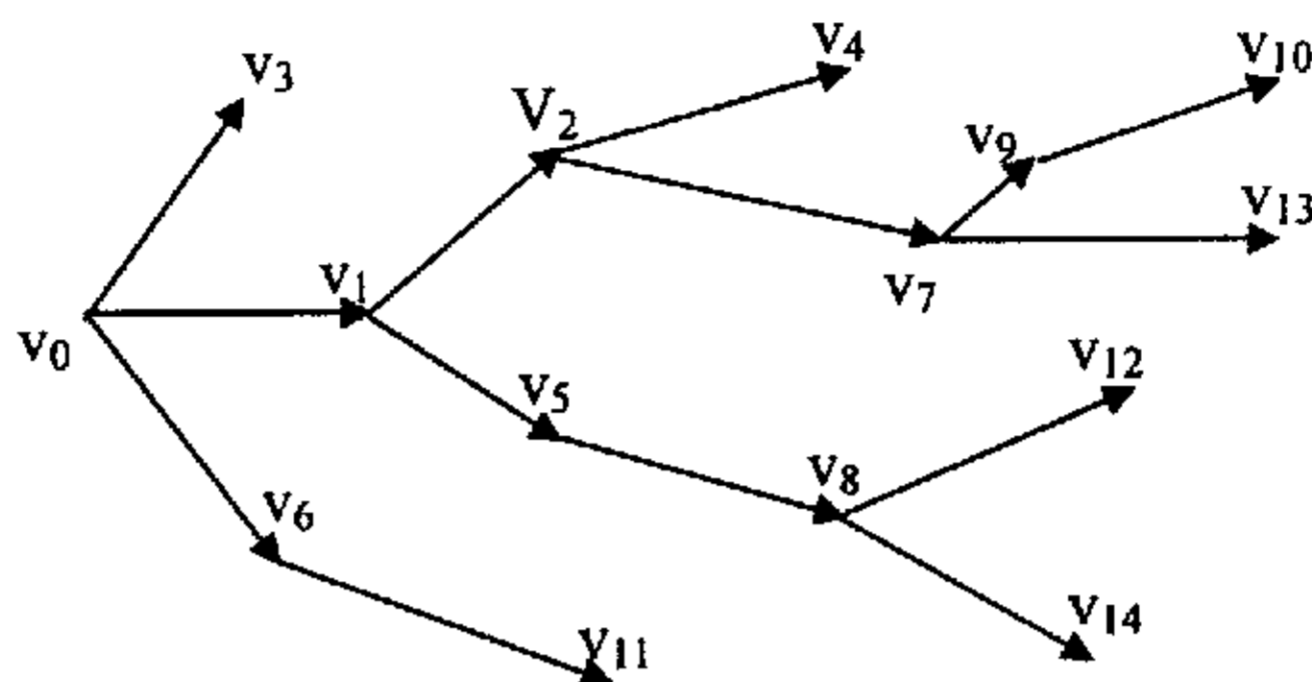


图 3-1 从树根到各结点的路径

根据以上分析我们可以得出求路径端点的算法。

设已求出从  $v_0$  出发长度依次递增的  $k$  条路径, 他们的端点各为  $v_1, v_2, \dots, v_k$ , 并归入已到达的端点集合  $T$  中, 并将  $v_0$  到  $v_i$  的距离简记为  $L(v_i)$ ,

对  $T$  外的任意结点  $v_i^-$  计算数值

$$L(v_i^-) = \min\{L(v_i) + w(v_i, v_i^-)\} \quad 0 \leq i \leq k \quad (3-5)$$

式(3-5)中约定  $L(v_0) = 0$ , 若  $(v_i, v_i^-) \notin E$ , 则令  $w(v_i, v_i^-) = \infty$ , (在计算机中则用某一允许的最大值代替  $\infty$ )。

因此, 第  $k+1$  条短路径, 其端点一定是  $T$  外各点中  $L(v)$  最小的那个点(这里已将  $v_0$  加入  $T$  中), 设为点  $v_{(k+1)}$ , 则

$$L(v_{(k+1)}) = \min\{L(v_i^-) | v_i^- \notin T\} \quad (3-6)$$

式(3-5)和(3-6)给出了求第  $k+1$  条短路径终点的公式, 然而直接用公式(3-5)

计算是很麻烦的，因为每求一条路径的终点，都要反复地对已通过的结点进行计算和比较，我们可以加以改进，即在求最短路径的过程中，不断修改未通过点的路径参数，于是得到如下算法

Dijkstra 最短路径算法

(1) for all  $v \neq u$   $L(v) \leftarrow w(u,v)$

(2)  $L(u) \leftarrow 0$

(3)  $T \leftarrow \{u\}$

(4) While  $T \neq V$  do

begin

(5) 找一点  $v_i^- \notin T$  且对所有  $v \notin T$   $L(v_i^-) \leq L(v)$

(6)  $T \leftarrow T \cup \{v_i^-\}$

(7) For all  $v \notin T$

If  $L(v) > L(v_i^-) + w(v_i^-, v)$

Then  $L(v) \leftarrow L(v) + w(v_i^-, v)$

End

算法说明

算法中指定结点设为  $u$ ,  $w(u,u)=0$ ,  $T$  为通过的结点集合，初始值为  $\{u\}$ ，当  $T=V$  是表示图的所有结点已通过，计算结束。计算从第四行循环语句开始，在不属于  $T$  的结点中选路径长度最小的一点  $v_i^-$ ，将  $v_i^-$  加入  $T$  中并根据  $v_i^-$  修改不属于  $T$  的结点的路径长度，如此反复进行直到所有结点都加入  $T$  中为止。因此这一算法是从指定点  $u$  到所有其余结点的距离，其值通过结点  $v_i^-$  的  $L(v_i^-)$  给出。如果只需要求  $u$  到某些点的距离，那么只要对第四行的循环语句稍作修改即可达到目的。这一算法的复杂性主要耗费在第四行开始的循环体中，而循环次数不会超过  $(n-1)$ ，在循环体中，第 5 行和第 7 行的执行时间是  $O(n)$  级的，总的运算时间是  $O(n*n)$  级的。



### 算法过程

每个节点从源节点沿已知最佳路径到本节点的代价或距离来标注(在括号内)。开始, 一条路线也不知道, 故所有节点都标注为 $\infty$ 。随着算法的进行和不断找到了路径, 标注也不断改变, 使之反映较好的路径。一个标注可以是暂时性的也可以是永久性的。当发现标注代表了从源节点到该节点的最短可能路径时, 就使它成为永久性的, 不再进行修改。

Dijkstra 算法在求只考虑路线的情况下, 被认为还是比较方便与快速的算法。由于一对一有配送业务一旦发生, 在配送量确定的情况下, 线路的安排是对成本的最大的影响。所以用此算法可行。

### 3.1.2 配送资源受限时的一对多配送方式分析

这一小节分析一配送中心在几个顾客之中对一种资源有限的产品分配的配送方案优化过程。在优化过程中考虑如何安排各时段的运输和库存计划才能最大限度地减少总的运输和库存成本。顾客的需求是随机的, 在做决策的时候, 库存成本和缺货成本都必须同直接运输成本一起考虑。

配送方案制订之初, 每个顾客的初始库存水平是已知的, 初始库存为前一天顾客处没有销售出去的货物数量。在配送过程中, 允许不给某些顾客送货, 每个顾客在一次配送过程中最多被访问一次。配送方案可以描述如下首先, 将各个顾客的初始库存(初始库存可能是前一天提供给每个顾客的货物所剩下的量)报告给配送中心, 这一信息用于确定第二天在各个顾客之间分配有效的产品。根据顾客的要求, 对各顾客的车辆及线路的分配同时也被确定。估计了交付量后(在一天结束的时候得到), 需求随之产生。同时在各个顾客中, 库存存货成本和缺货成本也同时发生了, 并与一天结束时的仓库存货水平成比例。

#### 3.1.2.1 建模与分析

对模型中的符号说明如下

$K$ —车辆数量

$n$ —顾客数量  $n=1, \dots, n$       0 代表配送中心

$d_{ij}$ —顾客  $i$  到顾客  $j$  的距离

$b_k$ —车辆  $k$  的运输能力(载重量或载容量)

$T_k$ —车辆  $k$  在配送中心的出发时间

---

$c_k$ —车辆  $k$  的单位距离行驶费用

$c_{ijk}$ —车辆  $k$  从顾客  $i$  到顾客  $j$  的直接运输成本  $c_{ijk} = c_k \cdot d_{ij}$

$r_{ik}$ —车辆  $k$  在顾客  $i$  处的作业时间, 设  $r_{0k} = 0$

$g_k$ —使用车辆  $k$  的固定费用

$v_k$ —车辆  $k$  的平均行驶速度 (km/h)

$t_{ijk}$ —表示车辆  $k$  从顾客  $i$  行驶到顾客  $j$  的平均时间  $t_{ijk} = d_{ij} / v_k$

$F_i(\cdot)$ —一个时期内在顾客  $i$  的需求量的累积配送货物函数, 为严格增长。

$hi^+$ —在顾客点  $i$  的每单位货物单位时间的存储成本

$hi^-$ —在顾客点  $i$  的每单位货物单位时间的缺货成本

$[e_i, l_i]$ —客户  $i$  的时间窗,  $e_i$  为车辆最早到达顾客  $i$  的时刻,  $l_i$  为其最晚到达顾客  $i$  的时刻。

$\beta_i$ —顾客  $i$  的初始库存量

$A$ —在配送中心的总的有效的配送产品的数量

有关顾客  $i$  处罚函数  $p_i$  如下

$$p_i(a_{ik}) = \begin{cases} f_{ie} + p_e(e_i - a_{ik}) & a_{ik} < e_i \\ 0 & e_i \leq a_{ik} \leq l_i \\ f_{il} + p_l(a_{ik} - l_i) & a_{ik} > l_i \end{cases}$$

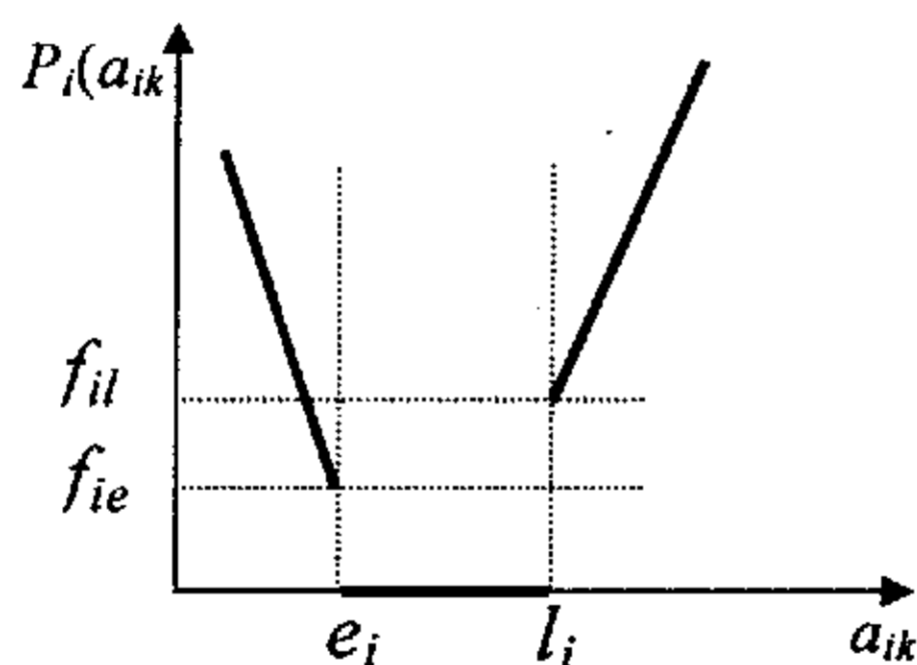


图 3-2 时间窗与惩罚函数关系图

$p_i(a_{ik})$ —为车辆  $k$  在时间  $a_{ik}$  服务顾客  $i$  所可能获得的惩罚成本

$a_{ik}$ —为车辆  $k$  实际到达顾客  $i$  时间

$$a_{ik} = t_{(i-1)ik} + a_{(i-1)k} + r_{(i-1)k} \quad a_{ik} = T_k + t_{0ik}$$

$f_{ie}, f_{il}$ —为车辆在顾客  $i$  早到和晚到的最低惩罚成本值, 即只要车辆违反顾客的时间窗, 就有保障顾客的最低惩罚成本

$p_e, p_l$ —代表早到或晚到时惩罚成本的正系数, 当顾客的时间窗宽度越窄

时其值越高，使违反程度越大时所获得的惩罚成本越高。

模型中的变量

首先定义一个虚拟路线  $k=0$ ，包括那由没有货物配送的顾客

$y_{ik}=1$  如果到顾客  $i$  车辆  $k$  被安排

0 其他

$x_{ijk}=1$  如果车辆  $k$  直接由顾客  $i$  到顾客  $j$

0 其他

$w_i$ —对顾客  $i$  的配送量

则库存成本函数  $q_i(\cdot)$  与它的派生函数  $q'_i(\cdot)$  可得出

$$q_i(w_i) = \int_{\beta_i+w_i}^{\infty} h_i^-(\xi - \beta_i - w_i) dF_i(\xi) + \int_0^{\beta_i+w_i} h_i^+(\beta_i - w_i - \xi) dF_i(\xi)$$

$$q'_i(w_i) = (h_i^+ + h_i^-)F_i(\beta_i + w_i) - h_i^- \quad i=1, \dots, n$$

这里，可以直接验证  $q_i(\cdot)$  是连续严格凸函数。

对建立该问题关于总配送成本最小的数学模型如下

$$(P) \min \sum_{i,j,k} (c_{ij} x_{ijk} + g_k) + \sum_i p_i(a_{ik}) + \sum_i q_i(w_i) \quad (3-7)$$

$$s.t. \quad \sum_i w_i y_{ik} \leq b_k \quad k=0, \dots, K \quad (3-8)$$

$$\sum_i w_i \leq A; \quad w_i \geq 0, \quad i=1, \dots, n \quad (3-9)$$

$$\sum_{k=1}^K y_{0k} = K \quad (3-10)$$

$$\sum_{k=0}^K y_{ik} = 1, \quad i=1, \dots, n$$

$$y_{ik} = 0 \text{ or } 1, \quad i=0, \dots, n; \quad k=0, \dots, K \quad (3-11)$$

$$\sum_i x_{ijk} = y_{jk}, \quad j=0, \dots, n; \quad k=1, \dots, K \quad (3-12)$$

$$\sum_j x_{ijk} = y_{ik}, \quad i=0, \dots, n; \quad k=1, \dots, K \quad (3-13)$$

$$\sum_{(i,j) \in S \times S} x_{ijk} \leq |S| - 1, \quad S \subseteq \{1, \dots, n\}, \quad (3-14)$$

$$2 \leq |S| \leq n-1; \quad k=0, \dots, K$$

$$x_{ijk} = 0 \text{ or } 1. \quad i = 0, \dots, n; \quad j = 0, \dots, n; \quad k = 0, \dots, K. \quad (3-15)$$

约束条件(3-8)(非线性约束)保证了每辆车被安排的载货量都在车辆的能力范围内。约束(3-9)保证了总的运输量在配送中心的有效配送货量之内。其它的约束条件的说明了同确定型 VRP 问题中所述。约束(3-10)和(3-11)保证每一个顾客只被安排了一条路线(有可能是虚拟路线)。约束条件(3-12)到(3-15)定义了经过安排路线上的顾客的旅行商问题(TSP)。

分析问题(P), 当确定了哪些车辆服务哪些顾客后(即  $y_{ik}$  的值被确定), 问题(P)可以被分解为两个问题——库存分配问题和  $K$  个 TSP 问题。

### 3.1.2.2 库存分配问题分析

#### 1 问题转化

当  $y_{ik}(i=1, \dots, n; k=0, \dots, K)$  的取值确定后, 就确定了对所有顾客的一个划分, 令  $Y_k$  表示车辆  $k$  服务的顾客集合, 这里  $Y_k = \{i: y_{ik}=1\}$ 。则库存配送问题可以写为如下<sup>[25]</sup>

$$(IA) \min \sum_{i=1}^n q_i(w_i) \quad (3-16)$$

$$\text{s.t.} \quad \sum_{i=1}^n w_i \leq A \quad (3-17)$$

$$\begin{aligned} \sum_{i \in Y_k} w_i &\leq b_k \quad k = 0, \dots, K \\ w_i &\geq 0 \quad i = 1, \dots, n \end{aligned} \quad (3-18)$$

设  $W_k = \sum_{i \in Y_k} w_i$ , 则问题转化为如下形式, 即对于每一个  $k=1, \dots, K$  有

$$\begin{aligned} Q_k(W_k) &= \min \sum_{i \in Y_k} q_i(w_i) \\ \text{s.t.} \quad \sum_{i \in Y_k} w_i &= W_k \\ w_i &\geq 0 \quad i \in Y_k \end{aligned} \quad (3-19)$$

等同于

$$\begin{aligned} \min \sum_{k=1}^K Q_k(W_k) \\ \text{s.t.} \quad \sum_{k=1}^K W_k &\leq A \quad 0 \leq W_k \leq b_k, \quad k = 1, \dots, K. \end{aligned} \quad (3-20)$$

对于这个非线性规划问题, 文献[29]称之为带广义上限的凸背包问题,

用一些典型的解带约束非线性规划问题的方法求解是比较繁琐和困难的。而采用 Bodin<sup>[30]</sup>设计的求解一类与  $IA$  问题有些相似的问题的算法来获得  $IA$  问题的近似解,相对求解更容易一些。为了论述清楚设计的求解过程,下面先简单介绍 Bodin 所研究的问题和算法。称他研究的问题为  $RA$ ,其目标函数如下:

$$\begin{aligned} & \min \sum_{i \in Y} f_i(w_i) \\ RA \quad & s.t. \quad \sum_i w_i = B; \quad 0 \leq w_i \leq s_i, \quad \forall i \end{aligned} \quad (3-21)$$

式中—— $f_i(w_i)$ 为连续可微的严格凸函数;

—— $B, s_i$ 是大于 0 的常数。

算法思路为

- 1)确定最优解中变量取值为 0 和取值为上界( $s_i$ )的两个集合;
- 2)确定与最优解对应的最优拉格朗日乘子, 然后根据它的取值确定最优解中各变量的取值。与本问题相关的一些过程描述如下

令  $w^*$  表示最优解向量,  $\lambda^*$  表示对应的最优拉格朗日乘子。定义  $I_0^* = \{i \in Y : w_i^* = 0\}$ ,  $I_p^* = \{i \in Y : 0 < w_i^* < s_i\}$ ,  $I_s^* = \{i \in Y : w_i^* = s_i\}$ 。由于  $f_i$  的一阶导数  $f_i'$  被假设为增, 同时根据非线性规划中的 Kuhn-Tucker 条件(最优性条件), 有如下一组关系式成立

$$\begin{cases} f_i'(0) \geq \lambda^* & i \in I_0^* \\ f_i'(0) < \lambda^* < f_i'(s_i) & i \in I_p^* \\ f_i'(s_i) \leq \lambda^* & i \in I_s^* \end{cases}$$

显然只有某个特定的对集合  $Y$  的划分才可能得到  $\{I_0, I_p, I_s\}$ , 为此, 定一个由  $f_i'$  在  $w_i=0$  和  $w_i=s_i$  的取值组成的集合  $R = \{[f_i'(0), 1], [f_i'(s_i), 0] | i \in Y\}$ , 令  $\geq_L$  表示  $R$  中元素的字典序关系,  $r^1$  表示集合  $r (r \in R)$  中的第一部分,  $r^2$  表示第二部分。如果将  $R$  划分成两个子集  $R_0$  和  $R_p$ , 并令  $d_0 = \text{lexmin } R_0, d_p = \text{lexmax } R_p$ , 那么有这样的关系成立

$$R_0 = \emptyset \quad \text{或} \quad d_p <_L d_0 \quad (3-22)$$

假设已经得到了  $R_0$  和  $R_p$ , 则相应的对  $Y$  的划分关系



$$\begin{cases} I_0 = \{i: [f'_i(0), 1] \in R_0\}, \\ I_p = \{i: [f'_i(s_i), 0] \in R_0, [f'_i(0), 1] \in R_p\}, \\ I_s = \{i: [f'_i(0), 0] \in R_p\}, \end{cases} \quad (3-23)$$

该文指出同时满足式(3-22)和(3-23)的对  $Y$  的划分中存在对应于问题  $RA$  的最优解的一个划分。当得到了这个最优划分  $\{I_0^*, I_p^*, I_s^*\}$  后, 他们给出了求解最优解的方法。对于任意的集合  $I \in Y$ , 定义函数  $G_I(\lambda) = \sum_{i \in I} (f'_i)^{-1}(\lambda)$ ,  $B_I = B - \sum_{i \in I} s_i$ 。则最优的乘子  $\lambda^*$  可以通过求解下式得到

$$G_{I_p^*} = B_{I_p^*} \quad (3-24)$$

当确定了  $\lambda^*$  后, 相应的有  $w_i^* = (f'_i)^{-1}(\lambda^*), i \in I_p^*$ 。从上面的公式可以看出  $G_I(\lambda)$  是由函数  $f'_i$  的一阶导数关于拉格朗日乘子的反函数构成的复合函数, 求解最优乘子的难易程度直接由函数  $f'_i$  的结构决定。具体到本章中的费用函数  $q_i$ , 因为它又包含随机需求量的分布函数, 所以在求解时需要一些处理技巧, 这将在介绍  $IA$  问题算法过程中给出。

基于 Bodin 的分析, Zipkin 设计了确定  $R_0$  和  $R_p$  的算法<sup>[31]</sup>。首先, 他证明了满足式(3-22)和(3-23)的一个划分  $\{I_0, I_p, I_s\}$  是最优划分的条件是当且仅当其至少满足如下两个关系式中的一个

$$[G_{I_p}(d_0^1), d_0^2] \geq L[B_{I_p}, 1] \quad (3-25)$$

$$[G_{I_p}(d_p^1), d_p^2] < L[B_{I_p}, 1] \quad (3-26)$$

然后他给出了确定  $R_0$  和  $R_p$  的几种方法, 如:

方法 1 从  $|R_p| = \emptyset$  开始, 逐步扩大这个集合的维数直到式(3-25)被满足;

方法 2 从  $R_p = R$  开始, 逐步减少  $R_p$  的规模直到式(3-26)被满足;

关于求解  $RA$  问题的详细分析和算法复杂性的讨论在文献[30,31]中有全面的论述。本节所关心的是如何扩展他们的思路来解决  $IA$  问题, 虽然这两个

问题的目标函数基本相同, 但约束条件相差的还是较大, 使得这个扩展过程变得比较困难, 下面根据本节问题的特点阐述所设计的求解过程。

## 2 算法过程

根据文献[25]中对此的求解过程, 库存分配问题的求解思路是, 首先分析约束式  $\sum_i w_i \leq A$  是不是原问题的紧约束, 如果不是, 可以将  $IA$  问题变形为类似  $RA$  的问题来解; 如果是, 先不考虑约束式(3-18), 根据  $RA$  问题的求解方法对约束式(3-17)进行适当的变化来解松弛掉约束(3-18)的  $IA$  问题; 然后依据初始解中确定的对客户划分(即确定的  $Y_k(k=1, \dots, K)$ ), 来判断前面所得的解是否满足约束式(3-18)。如不满足, 可解如下更易于用  $RA$  问题的方法解决的  $IAk$  问题

$$IAk \quad \begin{aligned} & \min \sum_{i \in Y_k} q_i(w_i) \\ & s.t. \quad \sum_{i \in Y_k} w_i = b_k \quad 0 \leq w_i, \quad i \in Y_k \end{aligned} \quad (3-27)$$

根据文献[25]中对约束(3-17)的分析及及  $RA$  问题的具体的改造过程, 利用其中所论述的变形  $RA$  算法, 在  $Y_k(k=0, \dots, K)$  已经确定的情况下,  $IA$  问题的具体求解过程如下

首先需要松弛掉约束(3-18), 就需要判断其是否是  $IA$  问题的紧约束。为此需要分析所有车辆的总配送能力与配送中心可提供的有限资源之间的关系, 分两种情况加以讨论

(1)  $\sum_k b_k \leq A$ , 则约束条件  $\sum_i w_i \leq A$  对最优解无影响, 可以不考虑。  $IA$

问题可以转化为如下形式

$$IA' \quad \begin{aligned} & \min \sum_{k=1}^K \sum_{i \in Y_k} q_i(w_i) \\ & s.t. \quad \sum_{i \in Y_k} w_i \leq b_k \quad \forall k; \quad w_i \geq 0 \quad \forall i \end{aligned} \quad (3-28)$$

由于  $\sum_k b_k \leq A$ ,  $IA'$  问题可以分解为相对  $K$  个车辆的  $K$  个子问题。如果解出与  $IAk$  问题相类似的  $K$  个  $IAk'$  (需要  $\sum_{i \in Y_k} w_i \leq b_k$ ) 问题, 则获得了  $IA$  问题的解。

求解  $IAk'$  的过程为

① 不考虑约束  $\sum_{i \in Y_k} w_i \leq b_k$ , 解  $IAk'$  获得一组变量  $w'_i (i \in Y_k)$ , 令

$$W_k = \min\{b_k, \sum_{i \in Y_k} w'_i\};$$

② 比较  $W_k$  与  $b_k$ 。如  $W_k < b_k$ , 取  $w_i^* = w'_i (i \in Y_k)$ ; 否则, 将松弛掉的不等式约束变为等式约束, 解  $IAk$  问题并将所求得解作为最优解。重复  $K$  次此过程便得到  $IA$  问题的解。

(2)  $\sum_k b_k > A$ , 则先松弛掉  $IA$  问题中的约束(3-18), 将问题转化为  $IAi$ ,

表示如下

$$IAi \min \sum_{i=1}^n q_i(w_i)$$

$$s.t. \quad \sum_{i=1}^n w_i \leq A \quad (3-29)$$

$$w_i \geq 0, \quad i = 1, \dots, n \quad (3-30)$$

比较  $RA$  问题, 与此问题类似, 可以采用  $RA$  问题的求解思路来解决此问题。求解  $IAi$  过程为:

① 不考虑约束(3-29), 求解  $IAi$  问题得  $\bar{w}_i (i = 1, \dots, n)$ ;

② 判断约束(3-29)是否被满足, 如果  $\sum_i \bar{w}_i \leq A$ , 取  $\hat{w}_i = \bar{w}_i (i = 1, \dots, n)$ , 同时

分析约束(3-18), 对于任意  $k (k \in K)$ , 如果  $\sum_{i \in Y_k} \hat{w}_i \leq b_k$ , 取  $w_i^* = \hat{w}_i (i \in Y_k)$ ;

如果  $\sum_{i \in Y_k} \hat{w}_i > b_k$ , 记  $k$  为  $k'$ , 采用文献[25]中的变形  $RA$  算法求解一个关于  $k'$

的  $IAk$  问题, 并将所得到的解作为最优解。如果  $\sum_i \bar{w}_i > A$ , 变  $IAi$  中约束(3-29)

为  $\sum_i \bar{w}_i = A$ , 并采用变形  $RA$  算法直接解此问题, 由于  $\sum_i \bar{w}_i > A$ , 说明配送

中的车辆能力有剩余, 所以可以通过运能调节得到一组可行解。并可以通过运用文献[25]中所设计的“贪婪”算法求得最优解。

### 3.1.2.3 一对多配送方案的线路确定

配送中心配送资源受限时的一对多配送在确定最后的配送方案时,就是获得一个最优的 VRP 问题的解。在确定配送方案时,在线路设计过程中并不是所有有需求的顾客都被安排在线路内,并且在改变服务于顾客的车辆安排的时候,配送总成本中除了运费变化外,库存费用也相应改变。

在配送中心配送资源有限的情况下对配送方案线路的确定,也是获得一个最优的 VRP 解,VRP 问题的算法有许多种,其中启发式算法近年来被应用广泛,其中的禁忌式搜索算法(Tabu Search, TS)自 Glover<sup>[32]</sup>提出以来,在各种领域被广泛应用,在 VRP 问题中应用 TS 算法的国外文献有许多,如 [33-36],都取得比较理想的解。本节中的问题 TS 启发式算法来确定最优车辆配送路线。

由于前面所有的讨论都是在假设  $y_{ik}(i=1, \dots, n; k=0, \dots, K)$  的取值确定的情况下进行的,所以在应用 TS 算法之前,可以应用文献[25]中对初始解的构造以及生成过程的 GENIUS 方法确定比较完善的初始解。然后可以采用 TS 算法获得最终满意解。

#### 1. TS 算法介绍

禁忌搜索算法最早是由 Glover 提出的,它是一种“局部搜索”的修正方法,是模拟人的思维的一种智能搜索算法,即人们对已搜索的地方不会立即去搜索,而去对其它地方进行搜索,若没有找到,可再搜索已去过的地方。禁忌搜索算法从一个初始可行解出发,选择一系列的特定搜索方向(移动)作为试探,试图找到更好的解,选择实现使目标函数值减少最多的移动,然后从这个新的解开始,继续寻找好的解,这一过程不断重复,直到找到的解不再改善为止。为了避免陷入局部最优解,禁忌搜索中采用了一种灵活的“记忆”技术,即对已经进行的优化过程进行记录和选择,指导下一步的搜索方向,这就是 tabu 表的建立。tabu 表中保存了最近若干次迭代过程中所实现的移动,凡是处于 tabu 表中的移动,在当前迭代过程中是不允许实现的,这样可以避免算法重新访问在最近若干次迭代过程中已经访问过的解群,从而防止了循环,帮助算法摆脱局部最优解。另外,为了尽可能不错过产生最优解的“移动”,禁忌搜索还采用“释放准则”的策略。

禁忌搜索算法中涉及的两个概念“移动”和“邻域”,移动始从一个解  $s'$  到另一个解  $s''$ ,“移动值”= $f(s')-f(s'')$ ,当移动值为负值时,移动得到改善。 $N(s)$ 是解  $s$  的邻域集合,  $N(s) \in S$ 。该算法在每一迭代步,对  $N(s)$ 内的所有移

动都进行检查,包括已改善的解和没有改善的解。为了避免盲目搜索,该算法用了一个所谓的“禁忌条件”,即移动是可接受的,否则,移动是禁忌的。禁忌只有在一定的迭代次数内是禁忌的。“期望条件”是为了保证某些“特殊”的移动。如果满足“期望条件”,禁忌移动也是可以接受的。也就是如果一个移动是特殊的,该算法是可以忽略其禁忌状态的。禁忌条件和期望条件就使得该算法具有从以前的迭代中获取信息的能力,对解空间的搜索具有智能性。在每次迭代中的可行移动中,接受移动值最小的移动,这一移动可能不是较好的,但它不会因为缺少改善的移动而失败,从而“跳出”局部最小值。禁忌移动是由“禁忌表”来控制的,其大小根据具体的问题来选取。“禁忌表”用来存放禁忌解,当禁忌表装满后,再来一个禁忌解时,就将表中的第一个禁忌解从表中移出,这时该解又是可移动的,如此不断更新。在每次迭代中,都可以得到一个最优解,这样就构成一个解序列,  $f^1, f^2, \dots, f^n$ , 选取其中最小者,即为所要寻找的最优解。主要步骤如下。

(1)产生初始解  $X_{int}$ ;

(2)设当前解  $X_{current}=X_{int}$ , 当前最好解  $X_{best}=X_{int}$ ;

(3)重复 3a~3f, 直到满足停止条件;

3a 在  $X_{int}$  的邻域内产生  $N_s$  个测试解  $X_i, 1 \leq i \leq N_s$ ;

3b 求出目标函数  $f(X_i)$ ;

3c 判断测试解是否在禁忌表中,若不在禁忌表或在禁忌表中但其目标函数值比  $X_{best}$  还好,则把它作为新的当前解  $X_{current}$ ,并转到 3d;否则,继续测试下一个测试解。若所有测试解都在禁忌表中,则转到 3a;

3d  $X_{best}=X_{current}$ ;

3e 若禁忌表已满,则按先进先出的原则更新禁忌表;

3f 把当前解  $X_{current}$  插入禁忌表;

(4)记下最优解  $X_{best}$ , 结束算法。

## 2. 具体算法设计

本问题的 TS 算法采用序列插入方法<sup>[37]</sup>构造邻域。计算初始,令  $X_{int}$  为当前解,即  $X_{current}=X_{int}$ ,  $m$  为上一过程产生的路线数。在当前解  $X_{current}$  中随机选择  $\min(n, 5m)$  个顶点,并依次将这些顶点插入到包含这些随机点当中五个最邻近顶点之一的路径中。在车辆服务的客户路径  $r$  中的顶点  $v$  被插入到路径  $r'$  中时,顶点  $v$  要从原路径中删除,并根据路径中顾客的改变重新为新生成的路径  $r$  和  $r'$  分派车辆,并重新计算变化后的顾客库存分配。然后对新



生成的解  $x'_{\text{current}}$  进行判断, 判断这个新解是否为  $X_{\text{int}}$  的领域内的最好解, 并求出此时相对应的函数值。同时判断新生成的解是否在 Tabu 表中, 如果不在表内, 且得到的函数值比移动之前还要优, 则设  $X_{\text{best}} = x'_{\text{current}}$ , 如果得到的目标函数值不优于移动之前的值, 则继续测试下一个解。并把  $X_{\text{current}}$  插入到 Tabu 表中, 并继续重复前面过程, 直至达到了规定的迭代次数的时候, 停止运算。

有关一对配送的算法过程在文献[25]中用具体的算例给予了计算, 算例分别以 35、50 和 75 个顾客为例, 在各个参数确定的情况下, 计算了最后的目标函数值以及车辆服务顾客的序列。均取得了较好的效果。

本节中的模型在对有关顾客时间窗的惩罚函数的处理上与文献[25]中所建的模型不同, 但是两个模型库存配送及处理上是相同的。所以用文献中所设计的算法也是合理的。

### 3.1.3 配送资源不受限时的一对多配送方式分析

在配送中心的配送资源不受限的情况下, 配送路线的安排配送可以照顾到所有有需求的顾客, 这时总配送成本中库存成本部分为一常数, 总配送成本不受顾客库存的影响, 因此库存成本不影响配送路线。此时建立关于配送总成本最小的数学模型如下

$$(P) \min \sum_{i,j,k} (c_{ij} x_{ijk} + g_k) + \sum_i p_i (a_{ik}) \quad (3-31)$$

约束条件为(3-8)、(3-10)—(3-15)

模型中的各参数说明同前。

这一模型也就是有时间窗的车辆调度问题(VRPTW), 观察模型本身, 可以发现, 如果只考虑约束条件(3-8)、(3-10)与(3-11), 则该模型为一分群问题。也就是说, 当某车辆  $k$  被指派去服务某一顾客群后, 其后续的问题为如何安排这一车辆的巡回路径, 以使成本最小化。所以, 原问题可以采取先分群(Clustering)再巡回(Routing)的求解策略, 将模型分解成分群主问题与  $K$ (车辆总数)个巡回子问题。因此模式分解如下

分群主问题

$$\min \sum_{k \in K} f(y_k) \quad (3-32)$$

约束条件(3-8) (3-10) (3-11)

其中  $f(y_k)$  为分群绩效指标, 即当某一顾客群被指派给车辆  $k$  后的绩效。

以  $N_k$  定义这一客户群, 则  $N_k = \{i | Y_{ik} = 1\}$ 。因此, 对每一车辆  $k$ ,  $f(Y_k)$  可定义一个巡回子问题

$$f(y_k) = \min \sum_{i \in N_k} \sum_{j \in N_k} (c_{ij} x_{ijk} + g_k) + \sum_{i \in N_k} p_i(a_{ik}) \quad (3-33)$$

$$\text{s.t.} \quad \sum_{i \in N_k} x_{ijk} = 1 \quad \forall j \in N_k \quad (3-34)$$

$$\sum_{j \in N_k} x_{ijk} = 1 \quad \forall i \in N_k \quad (3-35)$$

$$x_{ijk} = \{0, 1\}, \quad \forall i, j \in N_k \quad (3-36)$$

$p_i(a_{ik})$  规定同前。

通过如下分解分析可知, 要想获得  $\sum_{k \in K} f(y_k)$  必须先求得  $n$  个巡回子问题。

近年来, 对于 VRPTW 问题在进行理论研究时, 多用遗传算法求解。对上述二阶段求解过程, 对分群主问题可以设计一个遗传算法; 在求解巡回子问题部分, 可以采用节约启发式算法来求解。

用遗传算法求解式(3-32)问题, 可以采用如下步骤

步骤一 初始化如下各参数

车辆数  $K$ , 车辆的运输能力  $b_k$

顾客数  $n$ , 单个顾客的需求量  $w_i, i=1, 2, \dots, n$

编码(染色体与分群结果的相对应)

决定初始母体染色体数目  $s$ , 进化的迭代次数  $G$

交叉率  $p_c^*$ , 变异率  $p_m^*$

世代指针  $g=1$

步骤一的目的在于将有时间窗约束的车辆配送问题转换为遗传算法的形式, 并通过染色体编码的技巧, 以达到顾客分群的效果, 并产生巡回子问题。

在分群问题中, 决策变量为  $Y_i^k$ , 亦即第  $i$  个顾客是否被第  $k$  辆车所服务, 编码方式如下所述。假设配送中心的 3 辆车对 10 个顾客服务

顾客编号	1	2	3	4	5	6	7	8	9	10
染色体	1	2	1	3	1	3	2	3	2	3

则上图染色体所代表的意义为车辆 1 对顾客 1、3、5 服务；车辆 2 对顾客 2、7、9 服务；车辆 3 对顾客 4、6、8、10 服务。对顾客完成以上的分群后，只是对车辆服务的顾客信息清楚，对于每辆车的车辆路径如何安排才能使配送成本最小并不清楚，这就要求求解这三个独立的车辆的巡回子问题。

步骤二 随机产生  $s$  条染色体，每一条染色体代表一种分群结果，并淘汰违反车辆运输能力的染色体，对每一染色体存在如下判断

(1) 如果  $\sum_i w_i > b_k$ ，则淘汰该染色体，并重新随机产生一条染色体，进行

判断(2)；

(2) 如果  $\sum_i w_i \leq b_k$ ，则保留该染色体，直至母体染色体数为  $P$  时，则进行

步骤三。

步骤三

① 求解  $m$  个巡回子问题

巡回子问题是含有时间窗的旅行商问题，采取经过修正的 C-W 节约启发式算法，求解含有时间窗的巡回子问题。

以  $c_{ij}$  表示车辆从顾客  $i$  行驶到顾客  $j$  的费用，由 C-W 算法，得到顾客点  $i$  到顾客点  $j$  连接在一条线路上的费用节约值

$$s(i, j) = c_{i0} + c_{0j} - c_{ij}$$

然后按照费用节约值  $s(i, j)$  的大小连接点对。由于顾客对车辆服务的时间有约束，在决定车辆访问次序时，不仅要考虑行使距离的节约值，还要考虑等待损失和延迟成本，因此，需要对原费用节约值  $s(i, j)$  进行修正，得到

$$s'(i, j) = s(i, j) - \sum_{r \geq j} p_r(a_r) \quad (3-37)$$

式(3-37)中， $p_r(a_r)$  为连接顾客点  $i$  和顾客点  $j$  所在线路时，车辆在  $j$  后面的任务  $r$  处可能发生的时间惩罚成本。计算完所有  $s'(i, j)$  后，根据  $s'(i, j)$  大小连接点对。需要注意的是，每当连接一项后，各量可能发生变化，因此需要重新计算，在考虑连接，直至将该顾客群中的所有顾客安排到线路中。重复此操作，完成所有顾客群的巡回子路线问题。

② 计算各组染色体的适应值  $F_i$

$$F_i = \sum_{k \in K} f(y_k), \quad i = 1, 2, \dots, P$$

各车辆的适应函数值  $F_k$ , 各车的  $F_k$  加总求和后就得到各组染色体的适应函数值。经  $F_k(\text{转换后}) = F'_k / F_k(\text{转换后})$  ( $F'_k$  为初始群体中最佳染色体的适应函数值(转换前))转换后, 就得到转换后的适应函数值  $F_s$ , 转换的目的在于使用轮盘法复制时更加方便。

$$F_k = \sum_i \sum_j (c_{ij} x_{ijk} + g_k) + \sum_i p_i(a_{ik}) \quad (3-38)$$

$$F_k(\text{转换前的适应函数值}) = \sum_i \sum_j \sum_k c_{ij} x_{ijk} + \sum_i p_i(a_{ik}) \quad (3-39)$$

$$F_s(\text{转换后}) = F'_s / F_s(\text{转换前}) \quad (3-40)$$

其中  $F'_s$  为初始群体中最佳染色体的适应值。

#### 步骤四 染色体复制

以轮盘法复制  $s$  条染色体至交配盘。步骤如下

1. 计算每条染色体的适应函数值  $F_s$  的加总求和后得到  $F$ 。

$$F = \sum_1^s F_s \quad (3-41)$$

2. 计算各染色体的选择概率。

$$P_s = \frac{F_s}{F} \quad (3-42)$$

3. 计算各组染色体的累加概率。

$$q_s = \sum_1^s P_s \quad (3-43)$$

4. 随机产生  $p_r$

如果  $p_r \leq q_1$ , 则选择  $c_1$  染色体至交配盘;

如果  $q_{s-1} \leq p_r \leq q_s$ , 则选择  $c_s$  染色体至交配盘。

重复上述操作, 直至复制  $s$  条染色体至交配盘

#### 步骤五 染色体交叉

染色体交叉的目的在于通过染色体基因重组的方式, 产生新的染色体对, 本算法采用双点交换法的交叉方式, 步骤如下:

1. 从交配盘中随机挑选两条染色体;
2. 随机产生一个 0 到 1 的数值  $R$ ;
3. 比较交配率  $P_c$  与数值  $R$ , 判断所选的染色体对是否进行交叉如果  $R < P_c$ , 该染色体对进行交叉操作, 并进行步骤 4, 反之则进行步骤 5;

4. 随机产生两个交叉点,并对交叉点以内的两染色体基因进行交换重组;
5. 进行染色体变异操作, 并计算新的染色体的适应值。

#### 步骤六 染色体变异

进行染色体变异的目的在于通过染色体中数个基因特性的改变, 提高产生较佳子代染色体的机会, 本算法采用交换法的变异方式, 如下所述:

1. 在(0,1)范围内随机产生一个与染色体长度相等的数组

$$R=(r_1,r_2,\dots,r_n)$$

2. 比较突变率  $P_m$  与数组  $R$  中各个元素的大小, 判断所对应的基因座是否进行变异, 如果  $r_i < P_m$ , 则该染色体的第  $i$  个基因座进行变异, 否则保持原有基因不变。

3. 在配送车辆编号的范围内, 随机产生一个可代表配送车辆编号的数值, 此数值便为突变点上基因改变的目标。

#### 步骤七 染色体筛选

筛选染色体的目的在于通过约束式(3-8), 以检验经过进化后的染色体解是否违反单一车辆的载重量限制, 并将违反限制的染色体剔除, 以维持可行解的搜寻标准。

对每一条染色体有如下判断

如果  $\sum_i w_i > b_k$ , 则淘汰该染色体;

如果  $\sum_i w_i \leq b_k$ , 则保留该染色体, 并成为新的子代染色体。

#### 步骤八 检验是否达到深化停止条件

$$g = g + 1$$

如果  $g < G$ , 则返回步骤三;

如果  $g = G$ , 则进行步骤九。

步骤九 演算停止并选取新子代染色体群中适应度最好的染色体就是问题的较优解。

在文献[4]中的算例对这种改进遗传算法的有效性给予了充分的验证, 无论在运算速度与运算结果上都得到了比较满意的结果。

### 3.2 考虑配送中心库存多周期配送方案优化

配送中心在为顾客配送一般性的货物过程中, 为了满足顾客的需求量,



配送中心也需要维持一定量的库存, 这样, 不仅顾客本身发生了库存成本, 配送中心也同时发生库存成本, 这样, 在制订配送方案过程中, 不仅要考虑配送路线的安排, 也要考虑配送中心与顾客的库存成本。

这一小节分析了一配送中心对几个顾客的多货物品种配送方案优化过程。在优化过程中考虑如何安排各时段的运输和库存计划才能最大限度地减少总的运输和库存成本。

### 3.2.1 建模与分析

在对问题进行分析之前, 提出如下假设配送周期为离散的, 配送过程中顾客可以要求同时配送多品种货物; 在配送过程中配送中心库存量不受限制, 在周期内配送过程中顾客的需求量确定, 配送车辆型号相同并由配送中心统一控制。

对模型中的符号说明如下

$T$ —配送周期

$m$ —产品种类数量

$n$ —顾客数量  $n=1, \dots, n$

$O$ —配送中心

$d_{ij}$ —顾客  $i$  到顾客  $j$  的距离

$D_{ipt}$ —在  $t$  时刻顾客  $i$  对产品  $p$  的需求量

$b$ —车辆的运输能力(载重量或载容量)

$c$ —车辆的单位距离行驶费用

$c_{ij}$ —车辆从顾客  $i$  到顾客  $j$  的直接运输成本  $c_{ij}=c \cdot d_{ij}$

$g_t$ —在第  $t$  周期内使用车辆的固定成本

$O_p$ —配送中心订购货物  $p$  发生的固定成本

$h_{ip}$ —顾客  $i$  每周期内单位产品  $p$  的存储成本

$h_{op}$ —在配送中心  $O$  单位产品  $p$  的存储成本

$Z_{ipt}$ —在第  $t$  周期内顾客  $i$  产品  $p$  的库存量

$Z_{opt}$ —在第  $t$  周期内配送中心产品  $p$  的库存量

$w_{ipt}$ —配送中心在第  $t$  周期内为顾客  $i$  对  $p$  产品的的配送量

$Q_{pt}$ —第  $t$  周期内为配送中心配送的  $p$  产品的的配送量

模型中的变量

$y_{pt}=1$  如果在第  $t$  周期内配送中心为其库存补充产品  $p$ ,  $Q_{pt}>0$

0 其他

$x_{ijt}=1$  如果第  $t$  周期车辆直接由顾客  $i$  到顾客  $j(i \neq j)$

0 其他

$r_{it}$ —第  $t$  周期内配送中心向顾客  $i$  的配送次数

则模型的基本形式可以表示如下

$$P \quad \min \left\{ \sum_{i=0}^n \sum_{j=1}^n \sum_{t=1}^T (c_{ij} x_{ijt} + g_t) + \sum_{i=1}^n \sum_{p=1}^m \sum_{t=1}^T (h_{ip} Z_{jpt} + h_{op} Z_{opt}) + \sum_{p=1}^m \sum_{t=1}^T O_p y_{pt} \right\}$$

$$\text{s.t.} \quad Z_{opt} = Z_{op(t-1)} + Q_{pt} - \sum_{i=1}^n w_{ipt} \quad (3-44)$$

$$Z_{ipt} = Z_{ip(t-1)} + w_{ipt} - D_{ipt} \quad (3-45)$$

$$\sum_{\substack{i=1 \\ i \neq j}}^n x_{ijt} = r_{it} \quad i = 1, \dots, n; \forall t \quad (3-46)$$

$$\sum_{\substack{j=1 \\ j \neq i}}^n x_{ijt} = r_{it} \quad j = 1, \dots, n; \forall t$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijt} \geq \left\lceil \sum_{p \in S} \sum_{i \in S} w_{ipt} / b \right\rceil \quad \forall S \subseteq \{1, \dots, n\}; \forall t \quad (3-47)$$

$$y_{pt} \in \{0,1\} \quad x_{ijt} \in \{0,1\} \quad Q_{pt} \geq 0$$

$$Z_{opt} \geq 0 \quad Z_{j0} \geq 0 \quad Z_{jOT} \geq 0 \quad (3-48)$$

$$w_{ipt} \geq 0 \quad Z_{ipt} \geq 0 \quad r_{it} \geq 0 \text{ 且 } i = 1, \dots, n \quad \forall j, t$$

整个模型中，目标函数 P 表示总成本最小，包括运输成本、使用车辆的固定费用、顾客的库存成本以及配送中心的库存成本。约束(3-44)与(3-45)分别表示配送中心与顾客之间的库存平衡约束；约束(3-46)表示配送任务分配约束；约束(3-47)表示运输车辆装载能力约束，右边不等式表示按照配送数量与配送车辆装载能力计算的最低配送次数，也表示在使用的配送车辆完全相同的状况下，所表示所需要的车辆数；约束(3-48)表示变量非负约束。另外，由于模型中不考虑库存能力限制，故在模型中并未对配送中心的进货量进行限制，在周期内配送中心的进货量多少可以通过配送中心订货数量加以适当限定。

### 3.2.2 模型分解

上述模型 P 属于 NP 问题，可以分解为两个分别解决配送中心库存控制问题与运输调度问题的非线性规划模型。

模型 1 配送中心库存控制问题模型，该问题为无库存能力限制的多产品、多周期配送中心库存控制问题(IP)，以确定配送中心的最优补给数量和频率。模型如下

$$\text{IP} \quad \min \left\{ \sum_{p=1}^m \sum_{t=1}^T h_{op} Z_{opt} + \sum_{p=1}^m \sum_{t=1}^T O_p y_{pt} \right\} \quad (3-49)$$

$$\text{s.t.} \quad Z_{opt} = Z_{op(t-1)} + Q_{pt} - \sum_{i=1}^n w_{ipt} \quad (3-50)$$

$$\begin{aligned} y_{pt} \in \{0,1\} \quad Q_{pt} \geq 0 \\ Z_{opt} \geq 0 \quad Z_{j00} \geq 0 \quad Z_{j0T} \geq 0 \end{aligned} \quad (3-51)$$

模型 2 顾客库存分配问题(DP)，该问题与一般的 VRP 问题不同，在确定配送方案的时候，不仅仅要考虑配送货物种类、数量及优化路线，同时也要兼顾顾客的库存水平。由于原问题是多周期多货物品类的问题，所以这里假设每个周期内的配送是在周期末时候发生，并且配送车辆足够。模型如下

$$\text{DP} \quad \min \left\{ \sum_{i=0}^n \sum_{j=1}^n \sum_{t=1}^T (c_{ij} x_{ijt} + g_t) + \sum_{i=1}^n \sum_{p=1}^m \sum_{t=1}^T h_{ip} Z_{ipt} \right\} \quad (3-52)$$

$$\text{s.t.} \quad Z_{ipt} = Z_{ip(t-1)} + w_{ipt} - D_{ipt} \quad (3-53)$$

$$\sum_{\substack{i=1 \\ i \neq j}}^n x_{ijt} = r_{it} \quad i = 1, \dots, n; \forall t \quad (3-54)$$

$$\sum_{\substack{j=1 \\ j \neq i}}^n x_{ijt} = r_{it} \quad j = 1, \dots, n; \forall t$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijt} \geq \left\lceil \sum_{p \in S} \sum_{i \in S} w_{ipt} / b \right\rceil \quad \forall S \subseteq \{1, \dots, n\}; \forall t \quad (3-55)$$

$$\begin{aligned} x_{ijt} \in \{0,1\} \\ w_{ipt} \geq 0 \quad Z_{ipt} \geq 0 \quad r_{it} \geq 0 \text{ 且 } i = 1, \dots, n \quad \forall j, t \end{aligned} \quad (3-56)$$

### 3.2.3 算法设计

由于原问题是一个 NP 问题，问题复杂并且计算量大，通过直接求解很难得到模型的最优解，所以可以利用启发式算法，通过反复迭代，不断平衡配送中心与顾客之间对货物的补给需求的关系，不断协调之间的影响，不断优化策略，最终确定配送方案。在具体求解时，首先利用配送中心的库存控制问题(IP)的初始可行解解决整个对顾客的库存配送问题(DP)，这一过程被称为解基本问题(base case)。然后，分析顾客配送计划和车辆调度安排发生改变时，配送中心的订货决策可能受到的影响，即把综合考虑顾客配送计划和配送中心补给计划与分别确定顾客配送计划和配送中心库存补给计划的结果进行比较，用以分析综合考虑时的成本优势。当顾客与配送中心的库存补给计划都改变或是其中之一改变时，计算两者总成本减少的最大限度。反复进行这一过程，不断协调顾客配送计划与配送中心库存补给计划，直到总成本无法再减少为止。算法的主要过程包括三部分

#### 1. 解基本问题

通过对基本问题的求解，得到基本解，这一基本解就是在分别确定顾客配送计划与配送中心库存补给计划时得到的可行解。求可行解步骤如下

步骤一通过公式  $Q_{pi} = \sum_{i=1}^n D_{ipt}$  求解配送中心库存控制问题(IP)，使订货固定成本和配送中心库存存储成本之和最小。

步骤二运用步骤一计算得到的订货数量，即令  $w_{ipt} = D_{ipt}$  来解 DP 问题。

步骤三在不改变步骤一中得到的配送中心订货数量的前提下，通过计算，判断改变配送时间能否减少配送成本。如果可能，就用此结果代替步骤二计算得到的运输时间和运输成本。

在以后的计算中，将不再考虑配方案的确定对配送中心订货方案的影响。

#### 2. IP 问题和 DP 问题

对于这两个由模型 P 分解出来的模型，可以采用启发式算法确定每个周期内各种产品的配送中心的订货方案，以使周期内单位时间总成本最小。这种算法的实质就是要求得一个满足上述条件的  $t$  值，使  $T$  周期内配送中心  $p$  产品的库存补给数量等于  $k$  时期内对该产品的需求量，即需要满足

$Q_p = \sum_{k=1}^t D_{pk}$ 。设整个周期内发生的配送中心库存补充成本为  $TC(t)$ ，包括订货成本与库存存储成本。通过启发式算法不断逼近，搜索得到使  $TC(t)$  最小的  $t$  值。

启发式算法的基本思路是从  $t=1$  开始，逐渐增加  $t$  的值，直到  $TC(t)/(t) < TC(t+1)/(t+1)$ ，等到  $TC(t)/(t)$  的值。此时的  $t$  值一定可以满足配送中心对该产品的库存补充量的需求。 $(t+1)$  时期表示应进行新的库存补充订货的时间，重复上述迭代过程就可以求得对该产品的订货量  $Q_p$ 。

顾客库存分配问题(DP)的求解，主要是通过启发式算法来解决每个周期内的一系列车辆调度问题。由于单个顾客的需求可能大于车辆配送能力限制，因此需要配送的产品要分别装载在一整辆车和零担车上，故每个顾客的每个配送周期内所有产品的配送数量  $\sum_i w_{ip}$  可以分解成整车和零担车的装载量。

整车装载量可以由配送中心直接送至顾客手中；不足一车的零散货物量可以和其他顾客的需求再合并成整车进行联合配送，而联合配送的优化则为车辆调度问题，可以通过解 VRP 问题确定。假设各种产品占用配送车辆的装载能力(体积或重量)相同，每个配送周期内的配送成本等于直接配送成本、联合配送成本与顾客库存存储成本之和。这里，

直接配送成本 =  $\sum_i [($ 单车固定使用费用 +  $2c_{oi}) \times$  (为每个顾客  $i$  直接配送次数)]

联合配送成本 = 车辆运输成本 + (单车固定使用成本  $\times$  配送车辆数量)

VRP 问题可以通过改进旅行商问题(TSP)的算法来求解，首先给定几个初始解，然后不断逼近，选择其中使总成本最小的作为优化解。

在各周期内，配送中心在对多个顾客进行联合配送的情况下，各个顾客配送数量的初始解将通过扫描法(Sweep heuristic)确定。首先生成几个配送计划，将每个用户的配送计划作为种子点(seed point)，排除所有重复的配送路线，再利用三弧互换法(3-opt interchange)确定运输线路。其中，三弧互换法的基本思想是互换三条弧，如用另外三条弧代替运输路线中原有的三条弧，判断总成本是否降低，并保留能使总成本降低的线路。重复此过程，直到总成本无法降低为止，总成本最小时的可行解即为最优解。

### 3. 合并调整过程



合并调整过程主要是协调顾客库存补给与配送中心库存补给, 即通过改变某个顾客的本着计划, 分析其对顾客配送和配送中心库存补给可能带来的影响。这里主要是通过启发式算法调整向某顾客的配送时间, 进而合并配送数量, 利用整车运输以实现规模经济效益。假设在  $t$  时刻与  $t'$  时刻配送中心向顾客  $i$  配送  $p$  产品的货物量分别为  $w_{ipt}$  与  $w'_{ipt}$ , 经过上述调整后, 把  $t$  时刻向顾客  $i$  配送产品  $p$  的货物量转移到  $t'$  时刻, 即在  $t$  时刻不对顾客  $i$  配送货物  $p$ ,  $t'$  时刻对顾客  $i$  配送产品  $p$  的货物量增加到  $(w_{ipt} + w'_{ipt})$ 。经过这样的调整, 向顾客  $i$  配送货物  $p$  的整个配送方案就改变了, 同时导致运输成本的直接改变, 需要重新分析  $t$  时刻与  $t'$  时刻的车辆调度问题。

通过这种合并调整, 改变了配送方案, 进而改变了总成本。在顾客不允许缺货或延迟送货的情况下, 对所有  $t' < t$ , 当把  $t$  时刻的配送量转移到  $t'$  时刻时, 都会使运输成本降低。同时分析每次转移了配送量后, 包括配送中心库存补给与顾客配送在内的总成本是否减少。在实际运算过程中, 首选所有使运输成本减少的转换, 然后再考虑轮换后对配送中心库存成本的影响。

综上所述, 解决模型 P 的算法包括上述三大过程, 用以解决配送中心库存补给与顾客需求之间的影响。

## 第 4 章 配送方案优化的计算机实现

### 4.1 系统简介

#### 4.1.1 开发的必要性及意义

在配送中心的配送作业过程中,存在着大量的运筹和决策,配送中心配送方案优化问题就是亟待解决的一个重要问题。在配送作业中,管理者们需要根据顾客的具体需要采取有效的配送策略以提高服务水平,降低配送费用,而其中的配送车辆路线是否合理对配送的速度、成本、效益有着直接的影响。如果解决了车辆的线路优化特别是带有软硬时间窗的问题将有利于配送方案的优化制订,可以提高运输速度,达到降低配送成本,提高货物配送质量的目的,对企业经济效益有着重要的影响。

在第三章中论述的几种情况的模型建立与求解过程中,带有软硬时间窗的车辆路线的优化方案的实现是配送方案优化过程的关键部分。在车辆路线优化过程中可分为单车线路优化与多车线路优化,而单车线路的优化实现是实现多车线路优化的重要前提与基础,据此,笔者据此开发了一套解决单车货物配送方案管理的模拟系统。

#### 4.1.2 系统总体框架

本系统的最终目标是设计出根据输入的实验数据为配送中心实现货物配送提供优化的配送路线和提供参考的最优配送成本意见。系统总体框架见图 4-1 所示。

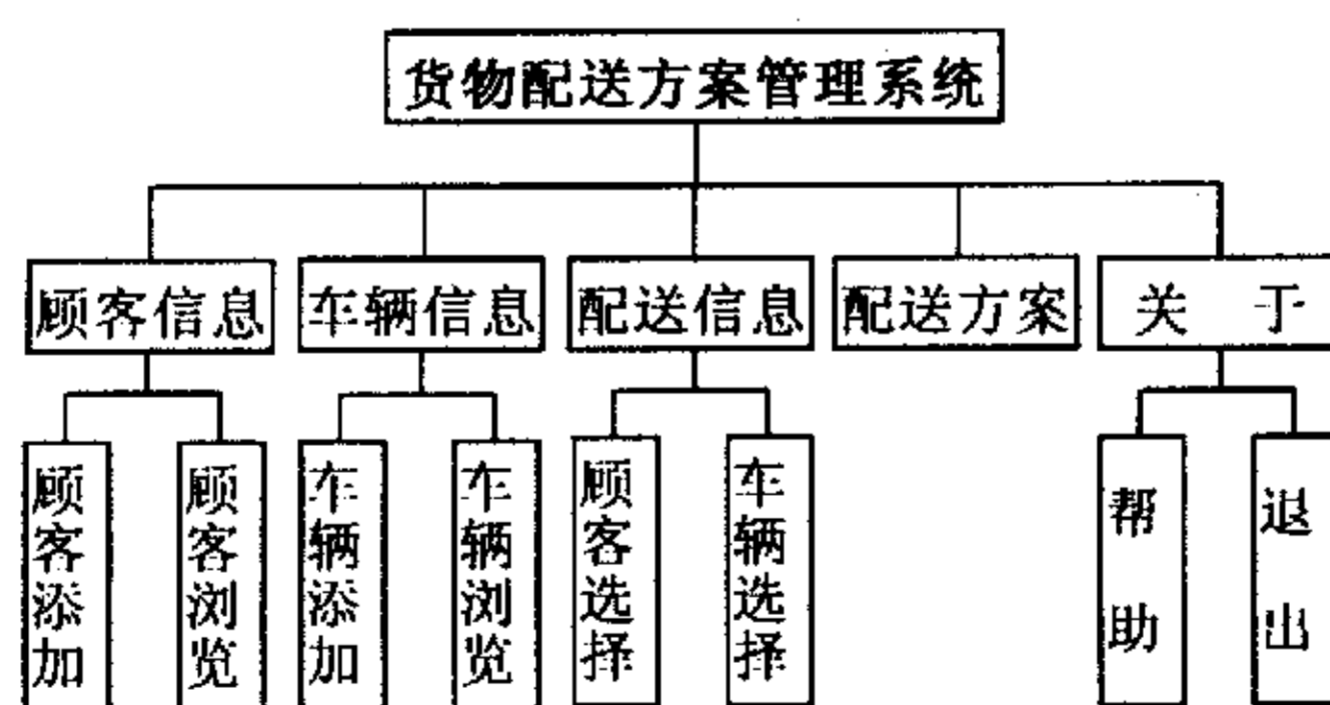


图 4-1 系统总体框架图

### 4.1.3 系统特点

本系统具有如下显著特点

1. 系统利用微软公司的 Access 的操作方便性与调用效果的优良性, 实现系统中的动态数据库的各种功能操作, 实现了 VB6.0 语言与 C 语言的有效集成, 并在 VB6.0 环境下调用 Access 数据库;
2. 系统包括三个模块顾客信息模块、车辆信息模块与配送信息模块;
3. 系统可根据需要输入顾客信息, 可任意修改删除顾客信息与各个顾客之间及配送中心与顾客之间的信息, 生成动态的基础数据库;
4. 由于顾客之间信息的不确定性, 结合动态数据库中生成的动态信息, 对顾客进行配送的优化后的车辆路线以一维线图显示出来;
5. 车辆路线显示的同时对此种配送方案的配送成本同时产生;
6. 本系统运算速度快, 可在较短时间内生成配送方案的优化显示。

## 4.2 系统软件的编制

### 4.2.1 系统软件开发环境

操作系统 WINDOWS 系列操作系统上运行, 并安装有 OFFICE2000 及以上版本办公软件, Microsoft Access 工作正常。

开发工具 Visual Basic 6.0, C 语言

### 4.2.2 数据库设计

当前关系数据库管理软件主要有 SQL Server, Oracle, FoxPro, Acces 等。其中 Oracle 和 SQL Server 适用于网络或大型数据库可开发。由于上述软件之间相互可以通过开放数据库互连(ODBC)技术进行数据访问, 所以数据库软件的选用对软件的升级不会有很大的影响。本课题选用 Access 数据库软件作为后台数据库。

Microsoft Access 是微软公司推出的在 Windows 环境下运行的关系数据库软件, Access 既是一个数据库管理系统, 同时又为用户开发应用程序提供了良好的环境。在中小型系统的开发中, Access 是能完全满足系统所需的功能。本系统的数据库设计见表 4-1——表 4-3。

表 4-1 Client 客户信息表

字段名	数据类型	是否可空	说明
ID	LONG INT	NOT NULL	记录号(主关键字)
NAME	CHAR(20)	NOT NULL	客户名称
CL_DE_QUANTITY	FLOAT	NULL	配送货物需求量(公斤)
CA_WO_TIME	FLOAT	NOT NULL	车辆作业时间(小时)
TI_WINDOW	CHAR(2)	NOT NULL	时间窗选择字段(默认无时间窗 00 硬时间窗 10 软时间窗 11)
EA_TIME	FLOAT	NULL	最早到达时间(时)
LA_TIME	FLOAT	NULL	最晚到达时间(时)
EA_LO_PU_COST	FLOAT	NULL	早到最低惩罚成本(元)
LA_LO_PU_COST	FLOAT	NULL	晚到最低惩罚成本(元)
EA_PU_COEFFIC	FLOAT	NULL	早到惩罚成本正系数
LA_PU_COEFFIC	FLOAT	NULL	晚到惩罚成本正系数
CEN_DISTANT	FLOAT	NOT NULL	与配送中心距离(公里)
CLI_CHOOSE	INT	NOT NULL	0 表示未选中, 1 表示选中(默认 0)

表 4-2 Distance 客户距离表

字段名	数据类型	是否可空	说明
TH_ID	LONG INT	NOT NULL	主客户记录号
ADJ_ID	LONG INT	NOT NULL	相邻客户关键字(外键)
AD_DIS	FLOAT	NOT NULL	与相邻客户之间的距离值(公里)
AD_SHORTEST	FLOAT	NOT NULL	与相邻客户间的最短距离(公里)
ROUTE_INDEX	CHAR(50)	NOT NULL	与相邻客户的最短路径

表 4-3 Car 车辆信息表

字段名	数据类型	是否可空	说明
ID	LONG INT	NOT NULL	记录号(主关键字)
CAR_NAME	CHAR(20)	NOT NULL	车辆名称
CAR_CAPACITY	INT	NOT NULL	车辆容量(公斤)
PER_AP_COST	FLOAT	NOT NULL	出一次车所发生的固定成本(元)
PER_ST_COST	FLOAT	NOT NULL	在顾客点停车一次的固定成本(元)
PER_AP_QUANTITY	INT	NOT NULL	出一次车的运输量(公斤)
AVE_RATE	FLOAT	NOT NULL	车辆平均速度(公里/小时)
PER_DIS_COST	FLOAT	NOT NULL	单位距离行驶费用(元/小时)
STA_TIME	FLOAT	NULL	车辆出发时间(小时)
CAR_CHOOSE	INT	NOT NULL	0 表示未选中, 1 表示选中(默认 0)

### 4.2.3 系统模块实现

系统主界面如图 4-2 所示。

图 4-2 中显示了系统总共包含三个功能模块, 即顾客信息模块、车辆信息模块与配送信息模块。系统主要模块功能如图 4-3 所示。

1. 顾客信息模块该模块主要完成顾客相关信息的录入、删除和修改。

该模块主要包含三项功能浏览顾客信息、增加顾客以及顾客信息编辑功能, 见图 4-4—图 4-6 所示。

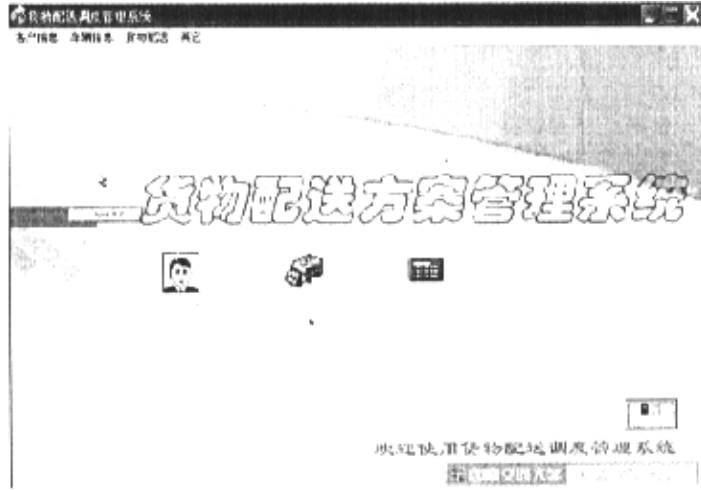


图 4-2 系统主界面

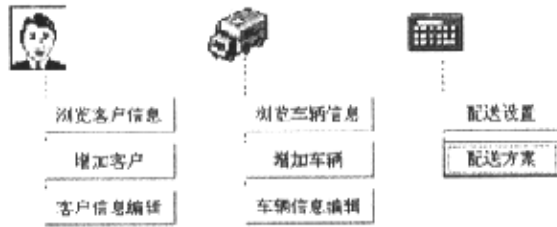


图 4-3 系统模块主要功能展开图



图 4-4 增加顾客信息对话框



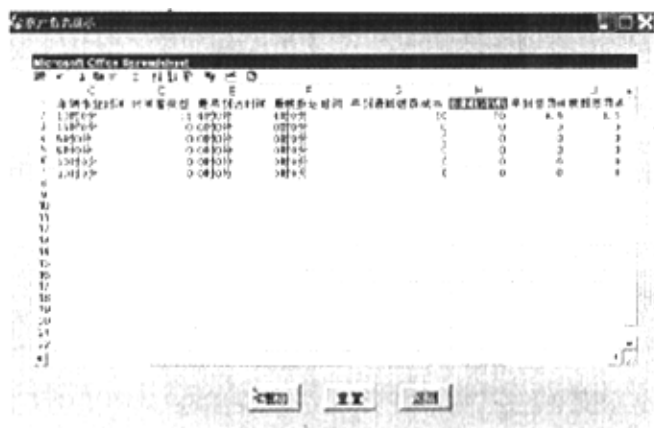


图 4-5 顾客信息浏览框

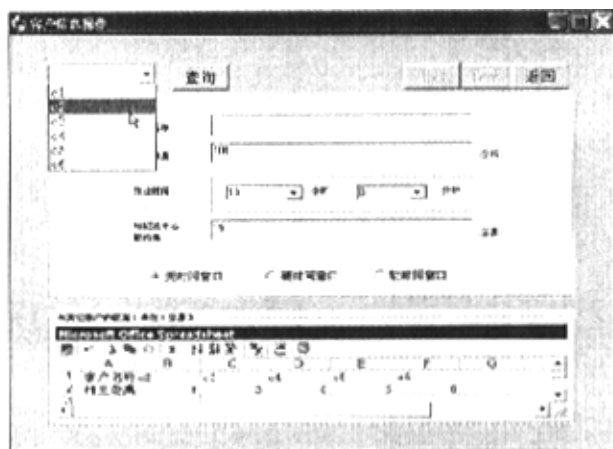


图 4-6 顾客信息编辑对话框

2. 车辆信息模块该模块主要完成车辆相关信息的录入、删除和修改。

该模块主要包含浏览车辆信息、增加车辆以及车辆信息编辑三项功能。

增加车辆菜单完成顾客信息的录入，当用户选择该菜单时，系统将弹出录入对话框，如图 4-7 所示；

浏览车辆信息菜单完成对已录入车辆信息进行显示，列出所有已录入车辆信息的简单信息(记录号、车辆名称)，如图 4-8 所示；

编辑车辆信息菜单完成对已录入车辆信息的修改删除等功能，如图 4-9 所示。

图 4-7 增加车辆信息对话框

车辆名称	载重量（公斤）	一次出车固定成本	一次下车固定成本	平均速度（公里/小时）	单位距离费用	车辆出发时间（小时）
t1	500	50	5	60	1.8	0分
t2	1000	60	10	20	1.3	5分
t3	1500	80	10	10	1.5	0分

对话框底部有‘添加’、‘重置’和‘返回’三个按钮。

图 4-8 车辆信息浏览框

3. 配送信息模块该模块完成在送货以前选择送货目的地、送货车辆以及相关信息的设置，为选择优化路径提供信息支持，如图 4-10 所示。

在这一功能模块中，如果所选的服务顾客的所有货物需求量超过所选车载重量，系统弹出标题为“车辆超重！”的对话框，并显示“车辆超重！请重新选择车辆！”对话框，并提示重新选择配送车辆。

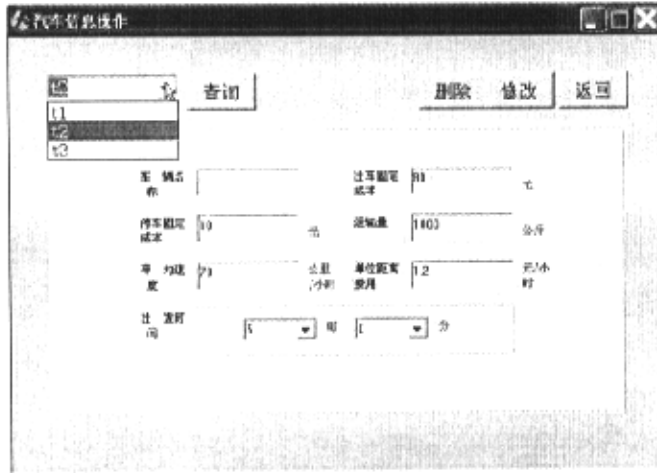


图 4-9 车辆信息编辑对话框

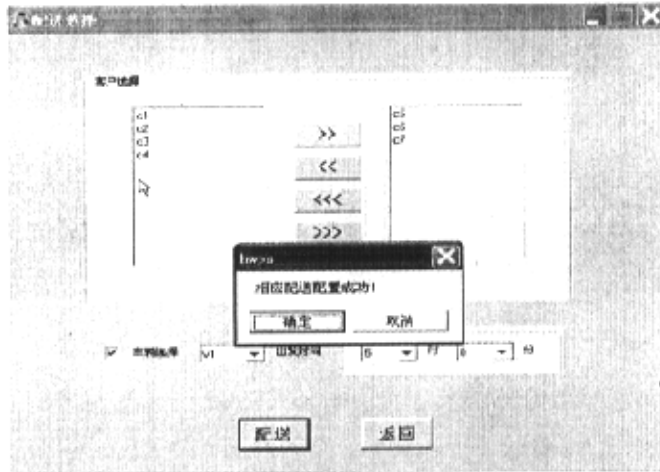


图 4-10 配送信息选择对话框

4. 计算成本模块该模块根据以上设定信息,计算最低成本并输出最低成本值,输出相应运输路线,如图 4-11 所示。

该配送最佳方案显示的含义为第一条一维直线表示从配送中心出发最后到返回配送中心的车辆经过所以所选顾客点的行走顺序,其下面的几条一维线表示车辆从配送中心出发经各组相邻的两顾客点间的车辆行走路线。如果两相邻被服务的顾客点之间的路径为最短时,车辆直接从前一顾客点到后一

顾客点，如果两相邻被服务的顾客点间的路径不存在或不为最短，车辆就会经过第三点从前一顾客点到达后一顾客点。

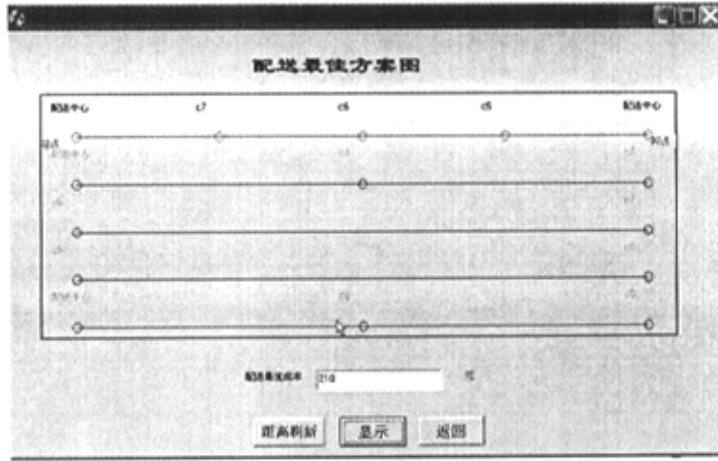


图 4-11 配送方案显示图

### 4.3 系统中关键模块功能实现流程

#### 1. 添加顾客信息

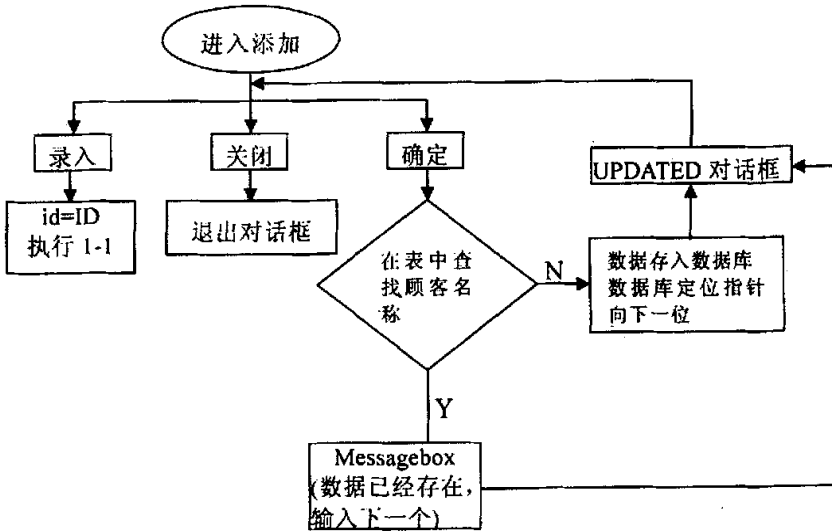


图 4-12 添加顾客信息流程图

## 1-1. 与顾客之间距离输入

当前信息录入顾客记录号 id,当前选定顾客记录号 id\_other,用户输入距离值 dista.

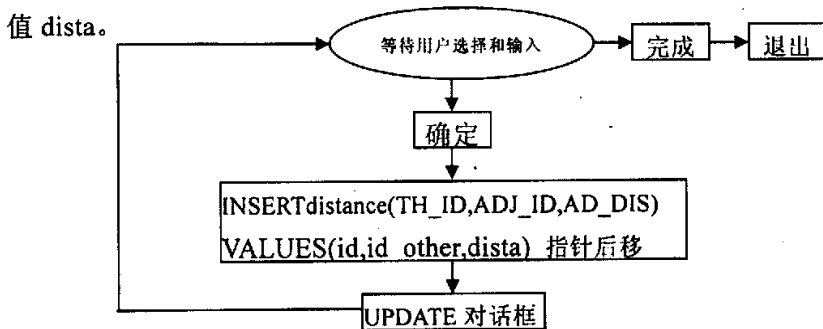


图 4-13 顾客之间距离输入流程图

## 2. 距离修改

当前信息修改顾客记录号 id,当前选定顾客记录号 id\_other,两顾客距离值 distance.

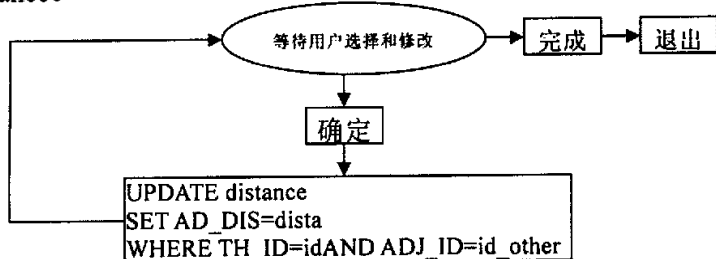


图 4-14 顾客之间距离修改流程图

## 3. 删除顾客信息

当前信息选定删除顾客记录号 id

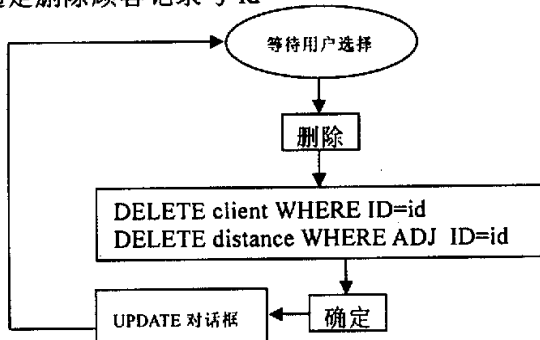


图 4-15 删除顾客信息流程图



4. 顾客需求

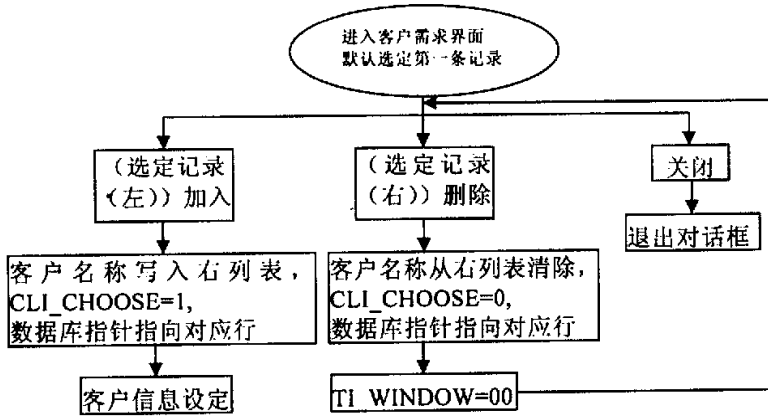


图 4-16 顾客需求流程图

5. 顾客信息设定

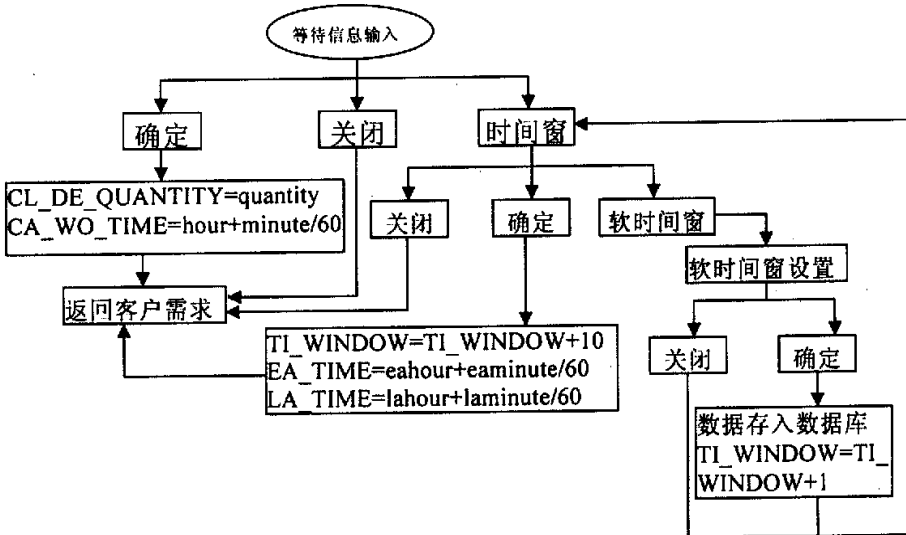


图 4-17 顾客信息设定流程图

## 6. 车辆选择

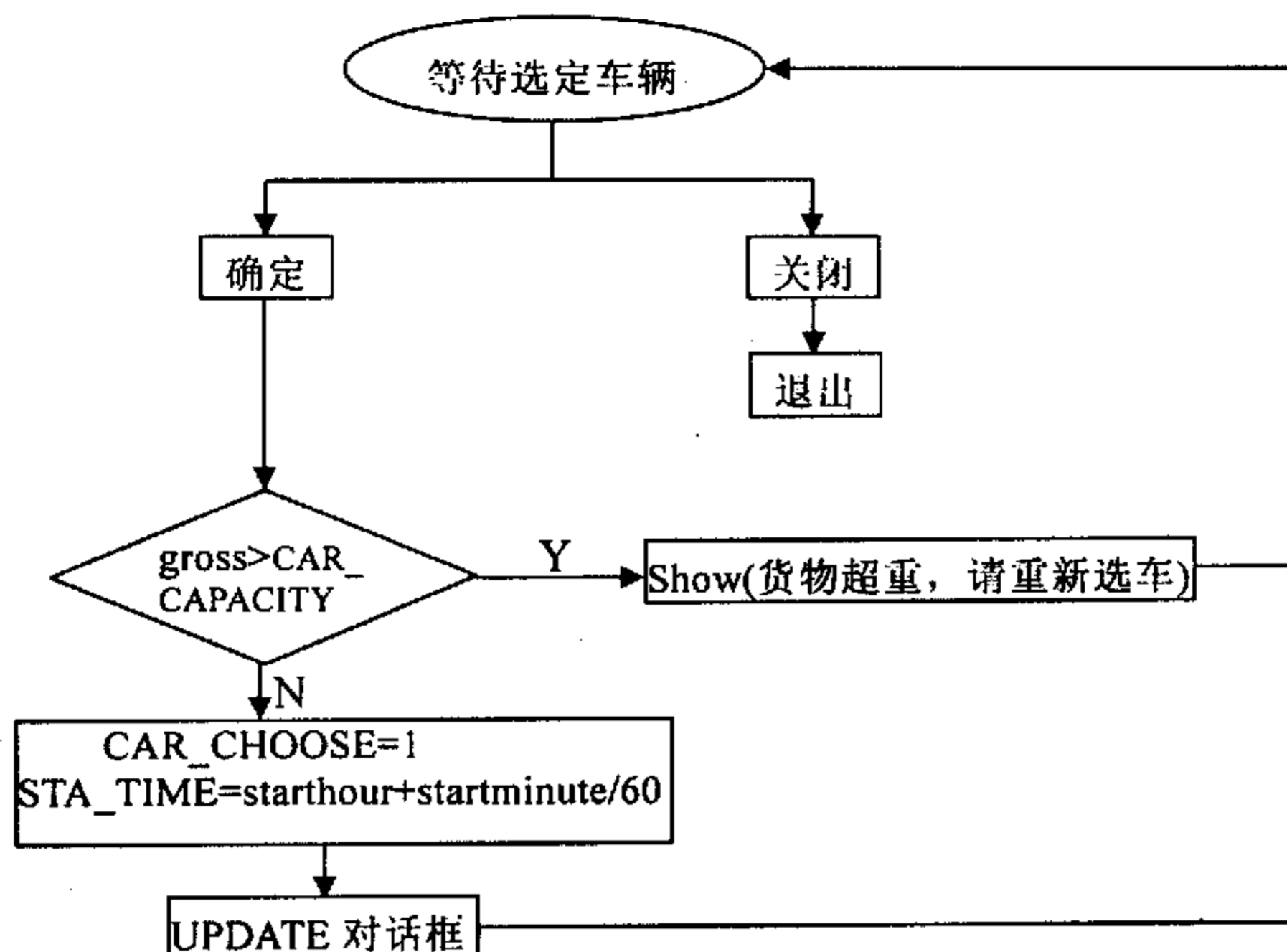


图 4-18 车辆选择流程图

## 4.4 开发多车系统需要解决的主要问题

以上有关系统特点及、软件编制以及系统中各个模块的实现都是基于单车配送方案优化的实现过程展开的论述。而在实际实用中,如在第三章中所分析的各种情况下的配送方案优化过程中,有关多车配送方案优化也是一个重要的问题,由于时间及计算机水平有限,这里只对多车配送方案优化系统在实现过程中必须解决好的相关问题进行简单论述:

1. 在配送中心的顾客数量确定的情况下,用于配送的车辆数量的确定,配送中心在为顾客配送过程中,在满足顾客需求的同时,最重要的目标就是追求总配送成本最小,而与总配送成本直接相关的车辆数量也就在配送服务发生之前首先需要确定;

2. 有关客户分组问题。无论配送中心的配送资源是否有限制,客户分组问题都将对多车配送方案优化产生着重要影响。合理的客户分组将直接决定配送中心用于配送的车辆数量的使用,也就直接影响着配送中心总配送成本;

3. 有关车辆选型问题。对于不同的顾客群需要用不同类型的车辆来服务,

如何为不同的顾客群选择配送车辆类型也影响着配送中心总配送成本；

4. 在配送中心配送资源有限的情况下，需要解决的一个重要问题就是配送中心的库存分配问题。由于配送中心的资源有限，并不是所有的顾客都能得到满足，同样的货物配送给不同的顾客在顾客处所产生的库存成本与运输成本各不相同，这时配送中心如何分配库存资源就会对最后的配送成本产生重要的影响。

## 结 论

本论文以分析配送中心配送方案制订的发展过程为前提, 研究了解决不同条件下的 VRP 问题的算法, 在此基础上, 在兼顾顾客对配送时间要求的同时, 将顾客对时间窗的要求与配送方案优化过程充分结合, 分别建立了在不考虑配送中心库存成本的单周期配送方案优化模型与在考虑配送中心库存成本的多周期配送方案优化模型, 并根据模型, 开发了考虑时间窗的单车配送方案优化管理系统。通过以上工作的完成, 得出以下结论

1. 在一对一的配送方案优化研究中, 当顾客需求量不确定时, 车辆满载的情况并不一定是最合理的; 但是当顾客需求量确定且有配送计划的时候; 配送中心配送方案的制订过程就转化为为配送车辆寻找最优路径的过程, 解决这一问题的最广泛算法是 DIJKSTRA 算法;

2. 在不考虑配送中心库存成本的单周期配送方案优化过程中, 如果配送中心的资源有限, 不是所有顾客的需求都能得到满足, 配送成本受顾客的库存影响, 其中的一个重要的问题, 即随机库存分配问题, 论文根据相关文献设计了用于解决这一问题的近似算法, 将禁忌搜索算法用到了这一问题中, 取得了较好的优化效果;

3. 如果配送资源不受限制, 即所有的顾客需求都可以得到满足, 顾客的库存成本不影响配送中心的最终配送成本, 问题就转化为带有时间窗的车辆路径问题(VRPTW), 将问题分解为两个阶段来设计求解, 在第一阶段使用遗传算法, 第二阶段使用快速有效的节约启发式算法。由于遗传算法其有快速求解与简化问题的优点, 所得到的结果可作为物流配送系统车辆路线决策的依据, 具有一定的实际应用价值;

4. 本论文在建立模型中所用的惩罚成本函数, 兼顾车辆在顾客时间窗限制之前与之后到达顾客位置, 并相应产生了配送车辆的的等待成本, 与车辆晚到顾客位置时所产生的惩罚成本, 符合实际应用, 能够达到较好的效果;

5. 在单车配送方案优化管理系统的开发中, 完成了几种模型中都涉及的核心问题, 即带有软硬时间窗的单车配送方案的确定。在开发过程中, 由于顾客信息的不确定性, 完成了建立动态基础数据库, 在配送方案优化过程中, 充分考虑了顾客对软硬时间窗的要求, 最终生成了车辆的最优配送方案一维显示与成本计算。该软件在 Windows2000 及以上环境下开发运行, 将

Visual Basic6.0 语言与 C 语言进行集成。

在配送中心进行货物配送工作中，配送方案优化问题是一个极具魅力的问题，优质的配送方案将产生极其可观的经济效益。本论文仅对配送方案优化过程进行了初步的研究，所获得的成果还很不完善，还有大量工作有待进一步去研究，主要体现在：

1. 本研究设定在单一物流中心、单一车种条件下进行配送服务，若能采用多个物流中心、多车种模式下进行排程，则配送方案优化过程将更有效率，而车辆的利用率也将相对的提高；

2. 实际的配送方案制订过程中，常有某些顾客必须优先服务，这就是具优先次序的配送方案优化问题，应以适当的模型加以表达。另外，装卸货物混合形态(Pick Up and Delivery)，将使得现实情况更加复杂，必须再以适当的模型表达；

3. 配送中心的车辆实际行驶于道路时，所面对的是复杂的交通状况，通常路段具有离峰、尖峰时间，尤其是以负责市区配送的配送中心最常碰到此类问题，因此配送时间并非固定值，车辆行驶成本也应随离、尖峰而有所不同，故应将实际的交通状况在模型中适当的加以考虑，以更能反应现实中的混合型时间窗限制车辆调度问题；

4. 配送方案优化过程中包含顾客分组、车辆分派、车型选择、配送货物品种多样化以及总体配送方案的拓扑显示等问题，这些问题在用计算机语言实现过程中都非常复杂，本论文中所开发的系统中并未涉及，将作为后续研究方向继续完成。

总之，本论文所做的研究工作是有意义的，但是仍有许多的研究工作有待进一步的开展和深化。

---

## 致 谢

本论文是在导师李宗平副教授的严格要求和精心指导下完成的。从论文的构思、开题，到论文的每一细节部分都凝聚着导师的心血。在研究生学习期间，李老师以其严谨的治学风格、渊博的学术知识和积极创新的生活态度，给予我莫大的教诲和启迪，这都将是我终生的财富。李老师在在学习上对我的严格要求与在生活上给予我的莫大的关怀和帮助，都深深地影响着我和感动着我，感激之情无以言表，谨在此表示衷心的感谢！

特别感谢马驹老师在我研究生的学习期间为我提供的宽松良好的学习环境以及在学习与生活上所给予的大量的关怀与指导，他的严谨的治学态度以及宽厚的人格魅力给我留下了深刻的印象，在此深表感谢；感谢严余松老师在学习生活中对我的悉心指导与无私帮助；感谢物流系叶怀珍教授、黄兴建主任在专业知识上的悉心指导与帮助。

感谢学院领导的大力支持与关怀，感谢资料室邹老师在我学习过程以及论文完成过程中所提供的大力支持与帮助，感谢学院研究生管理办公室杨永兰、陈艳等老师的支持与帮助。

感谢曲思源、戴英姿以及李德刚在我学生生活中所给予的特别的关心与帮助，感谢阎靖、陈玲玲、焦志莹、蒋赛、刘庆伟、张晓军、王艳等同学对我生活上的关心与帮助，是他们陪我度过了美好的研究生生活。

最后，要感谢我的父母以及爱人李晓兵，是他们给我的莫大的关爱支持着我，鼓励着我，使我能够不断前进，并得以顺利的完成学业。

赵冰洁

2004.5.24

---



## 参 考 文 献

- 1 李军. 有时间窗的车辆路线安排问题的启发式算法[J]. 系统工程. 1996, (5)
  - 2 马良. 旅行推销员问题的算法综述[J]. 数学的实践与认识. 2000, (4): 156~165
  - 3 徐宁, 李春光等. 几种现代优化算法的比较研究[J]. 系统工程与电子技术. 2002, (12): 100~103
  - 4 李军, 郭耀煌. 物流配送车辆优化调度理论与方法[M]. 中国物资出版社. 2001
  - 5 张运福, 时连强. 车辆路径问题(VRP)的一个启发式算法[J]. 河北建筑科技学院学报. 2002, (9): 81~84.
  - 6 李冰, 杜文. 单货物品种的运输和存储联合调度问题[J]. 西南交通大学学报. 2003, (12): 623~628.
  - 7 骆义. 物流配送车辆调度优化研究. 大连海事大学硕士学位论文. 2003, (3)
  - 8 汪祖柱, 程家兴. 求解组合优化问题的一种方法—分枝定界法[J]. 安徽大学学报(自然科学版). 2004, (1): 10~14
  - 9 张莹. 运筹学基础[M]. 北京:清华大学出版社. 1995
  - 10 Tim D.Nguyen. A neural network heuristic for the traveling salesman problem. B. S., University of Louisville Doctor thesis. 2002, (5): 2~16
  - 11 HuangBingxian. MemoryConsolidationandNeuralNetworkoftheHippocampus. Proc. IntConf Neu~ralNetworksandSignalProcessing. Guangzhou. 1993: 114~119
  - 12 Held, M.&R. Karp,(1970). "The Traveling salesman problem and minimum spanning trees". Operations Research. (18): 1138~1162
  - 13 Akaufmann. Integerand Mixed Programming: Theory and Application [M]. Academic Press, Inc(London), 1982: 95~106
  - 14 Solomon, M. M., 1983, "Vehicle routing and scheduling with time windows constraints: models and algorithms", Ph.D. dissertation, Dept of decision sciences, University of Pennsylvania
  - 15 Clarke, G.&Wright, J. W.. 1964, "Scheduling vehicles from a center delivery depot to a number of delivery points". Operation research quarterly. (12): 568~581
  - 16 Mole, R. H.&Jameson, S. R, 1976, "A sequential route~building algorithm employing generalized savings criterion". Operation Research quarterly. (27): 503~511
  - 17 Gillett, B.&Miller, L., 1974, "A heuristic algorithm for the vehicle dispatch problem",
-

- Operation research, (22): 340~349
- 18 Golden, B. L.&Skiscim, C. C., 1986, "Using simulated annealing to solve routing~ and location problems". *Maval research logistics quarterly*. (33): 261: 279
  - 19 Krolak, Felts, Marble, Nelson (1994), PC~based Interactive Decision Support System for Two~object Transportation problem. *Journal of Business Logistics*. 15(1)
  - 20 S. lin and B. Kernighan(1973), An Effective Heuristic Algorithm for me Traveling Salesman Problem. *Operation Research*. (21): 498~516
  - 21 N. Christofides, A. Mingozzi, and P.Toth, Exact Algorithms for the Vehicle Routing Problem, Based on Spanning Trees and Shortest Path Relaxations. *Math. Programme* to appear
  - 22 Gilbert Laporte, Helene Mercure and Yves Nobert(1992). A Branch and Bound Algorithm for a Class of Asymmetrical Vehicle Routing Problems. *J. Op1. Res. Soc.* 43, (5): 469~481
  - 23 Herron D.. Managing physical distribution for profit. *Harvard Business Review*, 1979, (79): 121~132
  - 24 Federgruen A., Zipkin P.. A combined vehicle routing and inventory allocation problem. *Perations Research*. 1984, 32, (5): 1019~1037
  - 25 袁庆达. 随机库存—运输联合优化问题研究. 西南交通大学博士学位论文. 2001, (12)
  - 26 A. Federgruen, P. Zipkin., A Combined Vehicle Routing And Inventory Allocation Problem. *Operations Research*. 32, (5): 1019~1036
  - 27 Federgruen A.& P. Zipkin. 1983. Solution Techniques for Some Allocation Problems. *Math. Program*. (25): 13~24
  - 28 袁庆达, 杜文. 带有软时间窗的混合车队车辆路线问题的模型与算法研究[J]. 西南交通大学学报. 2001, (4): 401~406.
  - 29 Bitran G., Hax A.. Disaggregation and resource allocation using convex knapsack problem with bounded variables. *Management Science*. 1981, (27): 431~441
  - 30 Bodin L.. Optimization procedures for the analysis of coherent structures. *IEEE Transaction on Reliability*. 1969, (R~18): 118~126
  - 31 Zipkin P.. Single ranking methods for allocation of one resource. *Mannagement Science*. 1980, (26): 4~43
  - 32 lover F..uture paths for integer programming and links to artificial
-

- intelligence. *Computer and Operations Research*. 1986, 5: 533~549.
- 33 Xu J., Kelly J P. A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science*. 1996, (30): 379~393
- 34 Hooker J.N., Natraj N.R.. Solving a general routing and scheduling problem by chain decomposition and tabu search. *Transportation Science*. 1995, (29): 30~44
- 35 Taillard E.. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*. 1997, (31): 170~186
- 36 Gendreau M, Hertz A., Laporte G.. A tabu search algorithm for the vehicle routing problem. *Management Science*. 1994, (40): 1276~1290
- 37 Gendreau, M.etc. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *computer&operations research*. 1999, (26): 1153~1173
- 38 B.Ombuki, M. Nakamura&M.Osamu. A Hybrid Search Based on Genetic Algorithms and Tabu Search for Vehicle Routing. presented at the 6th IASTED International Conference on Artificial Intelligence and Soft Computing, pp. 2002, (7) : 176~181
- 39 张丽萍, 柴跃廷, 曹瑞. 有时间窗车辆路径问题的改进遗传算法[J]. 计算机集成制造系统. 2002, (6): 451~454
- 40 冷德惠, 张金梅, 李大卫. 遗传算法在有时间窗车辆路径问题上的应用[J]. 鞍山钢铁学院学报. 2001, (3): 129~132
- 41 李大卫, 王莉, 王梦光. 遗传算法在有时间窗车辆路径问题上的应用[J]. 系统工程理论与实践. 1999, (8): 65~69
- 42 姜大立, 杨西龙, 杜文, 周贤伟. 车辆路径问题的遗传算法研究[J]. 系统工程理论与实践. 1999, (6): 40~45
- 43 P. Badeau, M.Gendreau, F. Guertin, J.~Y. Potvin and E. Taillard. A Parallel Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows. *Transportation Research*. 1997, (C 5): 109~122
- 44 刘民. 遗传算法方法及其在生产调度问题中的应用研究. [清华大学博士论文]. [TP]1997
- 45 李军, 谢秉磊, 郭强. 非满载车辆调度问题的遗传算法[J]. 系统工程理论方法应用. 2000, (3)
- 46 李军. 车辆调度问题的分派启发式算法[J]. 系统工程理论与实践. 1999, (1): 27~33
- 47 谢秉磊, 李军, 郭耀焯. 有时间窗的车辆调度问题的遗传算法[J]. 系统工程学报. 2000, (17)
-

- 
- 48 Balakrishnan, N., 1993, "Simple heuristics for the vehicle routing problems with soft time windows". *Journal of operational research society*. 44, (3): 279~287
  - 49 Desrocher, M., Desrosiers, J., &Solomon, M., 1992, "A new optimization algorithm for the vehicle routing problem with time windows". *Operations research*. 40, (2) : 342~354
  - 50 薛毅. 最优化原理与方法[M]. 北京: 北京工业大学出版社. 2001, (2)
  - 51 沈继红, 施久玉等. 数学建模[M]. 哈尔滨: 哈尔滨工程大学出版社. 1996, (5)
  - 52 L.R.富尔兹. 组合最优化[M]. 沈明刚等译. 上海翻译出版公司. 1988, (3)
  - 53 【美】D.P.柏塞克斯. 动态规划确定性和随机模型[M]. 李人厚等译. 西安: 西安交通大学出版社. 1990, (1)
  - 54 程理民, 吴江, 张玉林. 运筹学模型与方法教程[M]. 北京:清华大学出版社. 2000, (1)
  - 55 李宗平, 袁庆达. 基于卖方管理库存的配送决策优化模型与算法[J]. 西南交通大学学报. 2004, (2): 38~41
  - 56 吴璟莉. 配送中心车辆调度问题的研究. 广西大学硕士学位论文. 2003, (5)
  - 57 杨萍. 物流库存配送一体化优化研究与应用. 上海海运学院硕士学位论文. 2001, (12)
  - 58 李向阳. 遗传算法求解 VRP 问题[J]. 计算机工程与设计. 2004, (2): 271~276
  - 59 陈火根, 丁红钢, 程耀东. 物流配送中心车辆调度模型与遗传算法设计[J]. 浙江大学学报(工学版). 2003(9): 512~516
-

## 攻读硕士学位期间发表的论文 及参加的科研项目

### 1. 发表论文

- 1 赵冰洁. 铁路物流中心规划布局方案综合性能评价. 陕西工学院学报. 2004, (1): 57~60
- 2 赵冰洁, 马骊. 国有铁路进入城市交通的定位与发展模式. 铁道运输与经济. 2004, (4): 15~17

### 2. 参加的科研项目

- 1 四川省民航支线机场运营及旅游航线布局方案研究. 2002 年 12 月, 已完成
  - 2 福厦线 200km/h 客货混跑铁路车站间合理距离研究. 2003 年 4 月, 已完成
  - 3 提速线路车站平纵断面设计标准的研究. 2003 年 7 月, 已完成
  - 4 旅客列车运行成本和效益核算分析管理系统研究. 2003 年 12 月, 已完成
  - 5 铁路空重车调整模型与算法研究. 2003 年 10 月~至今 在研
-