学号 ： 2016302350068

密级 ：

武汉大学本科毕业论文

<基于细邻域禁忌搜索-变邻域搜索的周期性多车场车辆路径问题研究>

院(系)名 称 ： 经济与管理学院

专 业 名 称 ： 物流管理

学 生 姓 名 ： 邱俊彦

指 导 教 师 ： 汪挺松 副教授

二〇二〇年四月

郑 重 声 明

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名：邱俊彦　　日期：2020/5/15

摘　要

周期性多车场车辆路径问题(MDPVRP)是 VRP 问题的一个变种，在基本的 VRP 模型之上融合了多日期配送以及多车场配送的要素，一方面使得求解该类问题的难度较大，另一方面也使得此类问题与现代商业所面对的问题更为相似。总之，研究 MDPVRP 不仅对发现更有效的求解方法有理论意义，也对诸如材料供应、废物回收等商业业务有实际意义。

尽管 MDPVRP 于理论以及实际方面都有着一定的意义，但相比于 CVRP 和 VRPTW 等 VRP 问题而言，涉及该类问题的文献过少，所提出的解决方法亦相对较少，因此需要更多的求解方法来充实此类问题的研究。

本文针对 MDPVRP，提出了细邻域禁忌搜索-变邻域搜索的元启发式算法，并使用标准数据集对算法进行了检验，在较小规模的数据集上取得了可观的结果。又与优化求解器 GUROBI、当前最优算法(HGSADC 算法)进行了比较，从而总结出本文算法的优缺点。 本文所提出的新算法的主要内容如下：

(1) 依据“平摊不同日期的需求”设计了配送日期组合的细邻域搜索准则，依据“就近分配”原则设计了车场分配的细邻域搜索准则，并设置了用以控制细邻域大小的参数，从而可控地缩小搜索空间，加快搜索速度。

(2) 针对配送日期组合、车场的分配分别设定了禁忌搜索的要素。值得一提的是，禁忌表长度与车场数、车辆数、客户群规模、配送周期长短相关，使得禁忌表长度能与问题规模相适应，使得禁忌搜索在避免算法循环的条件下仍然能搜索到足够多的解。

(3) 针对每个车场的路径规划，采用结合 $2-opt^{*}$、$2-opt$ 和 *relocate* 三类邻域动作的变邻域搜索进行优化。

本文所提出的新算法的意义为：针对求解方法相对较少的 MDPVRP，为其提供一个有潜力的求解方向。

关键字：细邻域搜索；禁忌搜索；变邻域搜索；MDPVRP

Abstract

MDPVRP is a variant of VRP problem. It combines the periods and multi-depot on the base of VRP model, which makes it difficult to solve, and makes it more similar to the problems faced by modern business.In conclusion, the study of MDPVRP is not only of theoretical significance for finding more effective solutions, but also of practical significance for commercial businesses such as material supply and waste recovery.

Although MDPVRP has certain significance in both theory and practice, compared with CVRP and VRPTW, there are too few literatures related to this kind of problems and relatively few solutions proposed, so more solutions are needed to enrich the research of this kind of problems.

Aiming at MDPVRP, this thesis proposes a meta-heuristic algorithm of granular tabu search and variable neighborhood search, and tests the algorithm with standard data set, obtaining considerable results on a small data set.Compared with the GUROBI and current optimal algorithm (HGSADC algorithm), the advantages and disadvantages of this algorithm are summarized.

The main contents of the new algorithm proposed in this thesis are as follows:

(1) According to the "amortized requirements for different dates", the granular neighborhood search criteria for the combination of delivery dates is designed. The granular neighborhood search criteria for depot allocation is designed according to the principle of "nearby distribution", and the parameter to control the size of the granular neighborhood is set, so as to reduce the search space and speed up the search.

(2) Elements of tabu search are set respectively for the combination of delivery dates and the allocation of depot.It is worth mentioning that the tabu length is related to the number of depots, the number of vehicles, the size of the customer base and the length of the delivery horizon, so that the tabu length can be adapted to the size of the problem, so that tabu search can still search enough solutions without algorithm cycle.

(3) Aiming at planning the route for a single depot, variable neighborhood search combined with $2-opt^{*}$, $2-opt$ and $relocate$ is adopted for optimization.

The significance of the new algorithm proposed in this thesis is as follows: for

MDPVRP, which has relatively few solutions, it provides a potential direction of solving.

Key Words: granular neighborhood search; tabu search; variable neighborhood search；MDPVRP

目　录

第 1 章　绪论

1.1　选题背景及意义

车辆路径问题(Vehicle Routing Problem, VRP)是在诸如交通运输、供应链管理和生产计划等领域出现极多又极广的问题，自从 Dantzig and Ramser 针对货车调度问题提出了具有开创性的抽象模型和算法[1]，大量变种的 VRP 问题就被学者们提出并研究。其中，研究最深入的两个问题是考虑载重的 CVRP 和考虑时间窗的 VRPTW，与之相比，考虑周期性多车场的车辆路径问题(MDPVRP)并未获得充分多的研究，与之相关的文献和求解方法都相对较少。在有限的文献中，所涉及的求解方法大多是分阶段求解，并采用启发式或禁忌搜索算法进行优化，少有综合多种求解方法和概念的算法，因此 MDPVRP 在理论方面有待新方法的补充。随着商业活动业务的变化，垃圾回收[2]、维修服务[3]和材料供应[4]等业务需要考虑时间周期、多车场的因素来提高客户满意度以及降低成本，且此类业务需要求解速度快、稳健性好的算法，以应对动态变化着的市场需求。因此研究 MDPVRP 在现代商业方面也有着提升企业价值并降低运营成本的意义。

针对在理论与实际方面都有着研究价值的 MDPVRP 问题，本文提出了细邻域禁忌搜索-变邻域搜索(Granular Tabu Search and Variable Neighborhood Search, GTS-VNS)，在保持禁忌搜索全局寻优能力的同时，综合细邻域的概念加速搜索，并使用效果与效率兼备的变邻域搜索进一步加速搜索，以达到求解质量与求解速度的双重优化。

1.2　研究内容、方法及技术路线

1.2.1　研究内容

针对 MDPVRP 问题求解方法相对较少的现状，本文将提出细邻域禁忌搜索-变邻域搜索算法，以提供一个较有潜力的新方向。论文主要的内容为：

（1）绪论。该章首先从理论与实际方面介绍了选题背景及意义，通过考虑 MDPVRP 问题求解方法的不足，以及在现代商业方面日益增加的作用，从而体现

研究选题的意义。其次介绍了本文的研究方法及技术路线。

（2）文献综述。该章先从 CVRP、MDVRP 和 PVRP 的文献出发，介绍与之相关的算法，再介绍 MDPVRP 的有关文献，并指出此类问题现有的不足，并引申出本文所提出的新算法。

（3）相关概念及理论介绍。该章针对本文使用的细邻域禁忌搜索-变邻域搜索算法，分别介绍细邻域搜索，禁忌搜索和变邻域搜索的要素和流程，为合成出新算法奠定基础。

（4）MDPVRP 与细邻域禁忌搜索-变邻域搜索。该章先阐述 MDPVRP 问题的数学模型与分解方法，再说明新算法中各个部分的细节，最终给出用于解决 MDPVRP 问题的新算法的流程。

（5）标准数据集的优化。该章将细邻域禁忌搜索-变邻域搜索算法应用于 MDPVRP 问题的标准数据集，通过调节参数优化求解结果，并与已知最优解、GUROBI 和最优求解方法相比较，以说明新算法的潜力以及不足。

1.2.2 研究方法

（1）文献研究法。通过研读国内外对 MDPVRP 问题进行优化求解的文献，将经典方法进行适当的拓展，从而得到一种较有潜力的优化方法。

（2）元启发式算法。通过将细邻域搜索、禁忌搜索和变邻域搜索融合，提出用于解决 MDPVRP 问题的元启发式算法。

1.2.3 技术路线

本文技术路线如图 1.1 所示：

首先，通过对文献的介绍展现 MDPVRP 研究中的不足，并引申出本文的新算法。其次，为了介绍新算法，将分别介绍细邻域搜索、禁忌搜索以及变邻域搜索的概念及理论。然后，给出 MDPVRP 的数学模型，从而阐明最小化总路程的求解目标、车辆载重和车辆用时的约束。并针对 MDPVRP 问题的特性，提出独特的细邻域搜索准则、设定相应的禁忌搜索要素、设计完备的变邻域搜索，并给出用于解决 MDPVRP 的细邻域禁忌搜索-变邻域搜索算法的完整流程。最后，使用 MDPVRP 的标准数据集检验新算法，并将检验结果与 GUROBI、最优求解算法

(HGSADC 算法)进行对比，以说明新算法的潜力与不足。

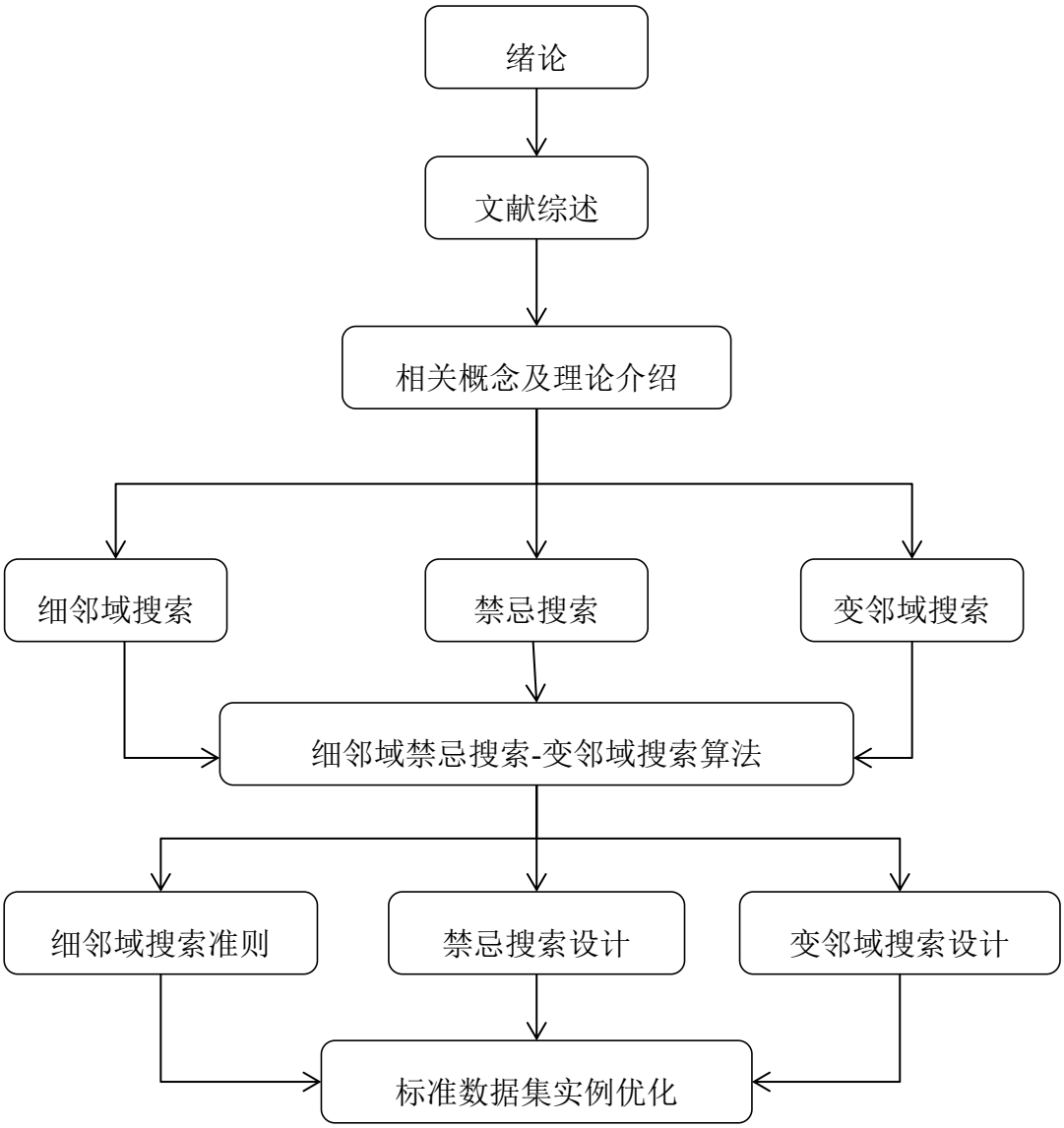

图 1.1 技术路线图

第 2 章　文献综述

考虑到多种 VRP 问题之间的内在联系性，该章将先介绍最基础的 CVRP，再介绍 MDVRP 以及 PVRP 的有关文献，最后介绍 MDPVRP 的有关文献。在介绍过程中，除了阐述文献中算法的思路，也会从算法的有效性方面进行适当补充。为了更全面地介绍文献，本章将分别从国外文献和国内文献进行介绍。

2.1　国外文献研究

CVRP 是最基础的 VRP 问题，围绕着该问题，已有众多的精确算法、启发式算法和元启发式算法被提出。在精确算法方面，Balinski and Quandt 提出了集合划分法，先计算出可行路径的集合，再从路径集合中选择可行路径[5]。Augerat et al 提出了分支剪枝法(branch-and-cut)，在分支定界法(branch-and-bound)的基础上，通过添加割平面的方式快速检验分支，从而加速了算法的收敛[6]，依据此算法，可解决的 CVRP 的客户数上限增加至 135 个。时至今日，分支剪枝法依然是十分优秀的精确算法。在启发式算法方面，Clarke and Wright 提出了 C-W 节约算法，依据两条路径结合产生的节约值的大小决定路线连接顺序[7]，相较于总是连接最近点的“最近连接法”，C-W 节约算法能给出相对不错的初始解。Gillett and Miller 提出了扫描法，先将客户群分组，使得每组都满足车辆载重约束，再对于每一组客户规划路径顺序[8]。与此同时，诸如路径间交换、路径内交换等改进算法不断地被提出，并体现了不错的优化效果，但遗憾的是，此类改进算法容易陷入局部最优。在元启发式算法方面，主要有 Glover 模拟“记忆功能”所提出的禁忌搜索算法[9]、Kirkpatrick et al 模拟“固体降温”所提出的模拟退火算法[10]、Holland 模拟自然选择与进化所提出的遗传算法[11]，这三类算法在实例优化中体现其优异的求解效果与效率，因此组成了元启发式算法的半壁江山。

MDVRP 的相关算法主要基于邻域搜索或者群体概念。在基于邻域搜索的算法中，Renaud et al[12] 以及 Cordeau et al[13]都使用了禁忌搜索算法，值得一提的是，Cordeau et al 通过统计邻域动作发生的频数来为多次发生的邻域动作添加罚成本，从而合理地增大了邻域搜索的广度，使得其提出的禁忌搜索算法一度成为 MDVRP 问题的最优求解方法。Lim and Zhu 使用了模拟退火算法，以概率的方式接受劣解，以此跳出局部最优[14]。Pisinger and Ropke 使用了自适应大邻域搜索，通过组合多

种 ruin-and-recreate 技术，极大地增加了邻域的多样性，从而极大地扩宽了邻域搜索的广度[15]，该算法也一度成为 MDVRP 问题的最优求解方法。在基于群体概念的算法中，Ombuki-Berman and Hanshar 使用了遗传算法，并针对车场的分配提出了独特的“突变”算子[16]，遗憾的是该算法的有效性不如基于邻域搜索的算法。Michel Gendreau et al 提出了自适应多样性控制的混合遗传算法(Hybrid Genetic Search with Adaptive Diversity Control , HGSADC)，在遗传算法的基础上，综合了邻域搜索与群体管理的方法，从而在保留遗传算法全局寻优能力的同时，混合了邻域搜索与群体管理跳出局部最优的特点[17]，使得该算法既有全局寻优能力，又有避免过早收敛的能力。在约 25%的标准数据集的优化中，HGSADC 算法刷新了已知最优解，在约 75%的标准数据集的优化中，HGSADC 算法与已知最优解持平，一举成为 MDVRP 的最优求解方法。

PVRP 的相关算法也主要基于邻域搜索或群体概念。在基于邻域搜索的算法中， Chao et al 首次采用接受劣解的邻域搜索算法[18]。 Cordeau et al 使用了禁忌搜索算法，并通过统计邻域动作发生的频数来调节将要使用的邻域动作[13]，使得其算法一度成为最优求解方法。Hemmelmayr et al 使用了变邻域搜索算法，并用三类邻域动作设计了变邻域[19]，值得一提的是，该算法的优化结果一举刷新了当时的已知最优解。在基于群体概念的算法中，Matos and Oliveira 使用了蚁群算法，通过该算法优化每个日期的配送路径[20]，可惜的是，该算法的有效性不如基于邻域搜索的算法。而 Michel Gendreau et al 提出的 HGSADC 算法[17]，不仅十分有效地适用于 MDVRP，也十分有效地适用于 PVRP。在约 50%的标准数据集优化中，HGSADC 算法刷新了已知最优解，在约 50%的标准数据集的优化中，HGSADC 算法与已知最优解持平，该算法亦因此成为 PVRP 的最优求解方法。

对于 MDPVRP，由于其所包含的因素较多，因此解决思路主要有两类，一类是同时决策运输日期、车场的分配与路径顺序，另一类是先决策运输日期或车场的分配，再决策路径顺序。在同时决策以上三类因素的研究中，Parthanadee and Logendran 提出了禁忌搜索算法，以用于解决允许延期交货的 MDPVRP 问题[21]，并在小规模算例中取得了不错的成果。Crainic et al 提出了用于解决多因素问题的综合合作搜索(Integrative Cooperative Search , ICS)[22]，并在考虑时间窗的 MDPVRP 算例中实现了优良的效果，但遗憾的是其未针对标准数据集给出相应的求解结果。Michel Gendreau et al 所提出的 HGSADC 算法[17]，不仅十分适合解决 MDVRP 与

PVRP，亦在 MDPVRP 标准数据集的实例优化中取得了十分优异的结果，在约 50%的标准数据集的优化中，HGSADC 算法的结果与已知最优解持平，在约 50%的标准数据集的优化中，HGSADC 算法的结果与已知最优解的差距在 0.5%以内，该算法也一举成为综合结果最优的求解算法。在分阶段决策配送日期、车场以及路径顺序的研究中，Hadjiconstantinou and Baldacci 先为客户群分配车场，再决策配送日期，最后再决策配送的路径顺序，并用禁忌搜索算法对车场和配送日期的分配进行优化[23]。Yang and Chu 先决策运输日期和车场的分配，再决策路径顺序，并用启发式算法优化运输日期和车场分配[24]。Kang et al 提出类似的方法，不同的是采用精确算法来优化运输日期和车场分配[25]。

2.2 国内文献研究

关于 CVRP，柏明国、李金书和韩梅提出了基于改进算法的启发式算法，通过将不可行路径上的节点穿插到可行路径的最优位置，从而优化目标函数[26]，可惜的是检验该启发式算法的算例过少，其有效程度仍有待检验。汪祖柱等学者使用了混合遗传算法，通过将遗传算法得到的子代进行邻域搜索，从而优化遗传算法的结果[27]，经标准数据集检验，该算法能以 99%以上的精度接近已知最优解，遗憾的是该算法的运行用时未被展示出来。张春苗、赵燕伟和冷龙龙使用了水波算法，通过定义“水波的传播”与“水波的折射”等算子，从而将该算法较好地应用于 CVRP 问题[28]，根据优化标准数据集的结果，该算法在 65%的数据集中能取得已知最优值，在 10%的数据集中能刷新最优值。周永务和彭碧涛考虑三维装载因素，并提出相应的禁忌搜索算法进行求解[29]，不足的是该算法的优越性未得到证明。除了以上较为典型的文献，在基本的元启发式算法基础上，我国学者提出了诸如改进蚁群算法、改进遗传算法等许许多多的改进算法，也提出了诸如混合蚁群算法、混合遗传算法等许许多多的混合算法，并在一定程度上取得了更优的效果。

关于 MDVRP，戴树贵等学者使用了混合蚁群算法，主要是在基本的蚁群算法之上，对得到的解进行 2-opt 优化，使得算法收敛速度和收敛效果得到提升[30]。党立伟和孙小明使用了混合遗传算法，通过对遗传算法得到的解进行邻域搜索，从而优化求解结果[31]，不足的是算例所用的数据集过少，无法有效证明算法的优越性。戚远航等学者提出泰森多边形的离散蝙蝠算法，通过泰森多边形加速搜索，

并通过蝙蝠群体进行寻优[32]，在约 50%的标准数据集的优化中，该算法的优化结果能与已知最优解持平。在算法层面，我国学者针对 MDVRP 所使用的算法相对较少，大多使用的是遗传算法或蚁群算法。不过，我国学者针对 MDVRP 的应用研究得较多，曾正洋等学者考虑紧急情况下的多车场运输，使用了变邻域下降法进行求解[33]，从而为紧急情况下的物流决策提供了一个可行的模板。薄非凡和魏法杰考虑城市垃圾清运，将之抽象为一个 MDVRP 问题，并使用混合遗传算法求解[34]。除此之外，我国学者通过结合实际因素，提出了 MDVRP 的许多应用方式，大大拓广了 MDVRP 的实际价值。

关于 PVRP，齐桐萱针对废旧家电，考虑周期性的回收方案，从而抽象出 PVRP 问题，并使用禁忌搜索优化该问题[35]。刘沙针对医疗废物回收，提出了分日期回收的方案，从而抽象出 PVRP 的问题，并使用混合遗传算法进行求解[36]。蔡婉君等学者在蚁群算法的基础上，对蚁群算法得到的解进行局部搜索，从而优化结果[37]。针对 PVRP 问题，我国学者提出的文献相对较少，且大多是根据实际因素抽象出 PVRP 进行求解，所提出的新算法相对少。

而关于 MDPVRP，我国学者提出的文献十分稀少，在中国知网上几乎找不到对应文献。可见，MDPVRP 的求解方法依然有待补充。

综合国外文献和国内文献的研究，相比于 CVRP、MDVRP 与 PVRP，用于解决 MDPVRP 问题的方法相对较少，且综合多种求解技术的方法更为稀少。本文将综合细邻域搜索、禁忌搜索和变邻域搜索的求解技术，提出细邻域禁忌搜索-变邻域搜索算法，以提供一个有潜力的求解方法。

第 3 章　相关概念及理论介绍

针对本文将提出的细邻域禁忌搜索-变邻域搜索算法，本章将介绍与之有关的细邻域搜索、禁忌搜索以及变邻域搜索。

3.1　细邻域搜索

3.1.1　细邻域搜索的内涵

细邻域搜索是在邻域的基础上，不去搜索一些不太可能得到最优解的邻域解，从而减小搜索空间，让邻域搜索的速度加快。设计细邻域搜索的核心就是设计邻域解的评估函数与临界值，只有评估值小于临界值的邻域解才会被搜索。Toth and Vigo 针对带载重的车辆路径问题(Capacitated Vehicle Routing Problem, CVRP)，选择要移动的弧的长度作为邻域动作的评估函数，选择弧的平均长度 $\tilde{d}$ 与常数 β 的积作为临界值[38]，如公式 3.1 所示：

$$\begin{aligned} \tilde{d} &= \cos t(S^*)/(n+K) \\ threshold &= \beta \cdot \tilde{d} \end{aligned} \tag{3.1}$$

公式 3.1 中，S^* 表示已知最优解，$\cos t(S^*)$ 表示已知最优解的弧的总长，n 表示客户节点数，K 表示车辆数，β 表示一个常数。如果邻域动作所连接的弧的长度大于 $threshold$ ，则该邻域动作将不被搜索。可见，细邻域评估函数与临界值的设定方式与所处理问题的属性相关，本文将依据 MDPVRP 的特性提出独特的评估函数。

3.1.2　细邻域搜索的流程

表 3.1　符号说明

符号	说明
S	问题的解
$S_{current}$	当前解
S^*	已知最优解
$threshold$	用于判断邻域解是否被搜索的临界值
$\cos t(S)$	解对应的成本值
$G(S)$	解 S 的评估函数
$N(S_{current})$	当前解的邻域空间

细邻域搜索的流程为：

(1) 生成初始解作为当前解 $S_{current}$

(2) 从当前解的邻域 $N(S_{current})$ 中选择一个邻域解 S 并计算其评估函数值 $G(S)$。

(3) 如果评估函数值 $G(S)$ 大于人为设定的临界值 *threshold*，则返回第 2 步，并搜索下一个邻域解。如果小于临界值 *threshold*，则运行第 4 步。

(4) 判断邻域解 S 的成本 $\cos t(S)$ 是否优于已知最优解的成本 $\cos t(S^*)$，如果优于，则将解 S 赋值给已知最优解 S^*。如果不优于，则返回第 2 步并搜索下一个邻域解。

(5) 将邻域搜索得到的具有最小成本的解 S 赋值给当前解 $S_{current}$，检验迭代次数是否达到上界，若达到，则算法结束，若未达到，继续返回至第 2 步。

细邻域搜索的伪代码如表 3.2 所示。

表 3.2 细邻域搜索的伪代码

Algorithm 3.1: Granular Neighborhood Search

1. 生成初始解 $S_{current}$
2. **While** *the stopping criterion is not met* **do**
3. **For** $S \in N(S_{current})$
4. **If** $G(S) > threshold$ **then**
5. **continue**
6. **If** $\cos t(S) < \cos t(S^*)$ **then**
7. $S^* := S$
8. $S_{current} := S$,**where** $\cos t(S)$ is minimal

3.2 禁忌搜索

3.2.1 禁忌搜索的要素

Glover 所提出的禁忌搜索(Tabu Search, TS)，是基于邻域搜索并模拟“记忆功能”的智能算法[9]。其基本思想是选择当前解 $S_{current}$ 的邻域空间 $N(S_{current})$ 中最好的解作为新的当前解，从而跳出局部最优，并将已遍历的解记录到禁忌表中不予再次访问，从而避免算法的循环。

综上，禁忌搜索的基本要素如表 3.3 所示。

表 3.3　禁忌搜索的基本要素

要素	作用
邻域动作	搜索更多的解
禁忌对象	记录已遍历的解或邻域动作，并禁止再次访问
禁忌表长度	禁忌对象被禁止访问的期限
破禁准则	解禁会优化已知最优解的禁忌对象

3.2.2　禁忌搜索的流程

禁忌搜索的流程为：

(1) 生成空禁忌表，生成初始解作为当前解 $S_{current}$，并将 $S_{current}$ 记录至禁忌表。

(2) 从当前解的邻域 $N(S_{current})$ 选择邻域解 S，如果其已在禁忌表中，则检验破禁准则，若不符合破禁准则，则重新运行第 2 步。

(3) 判断 S 的成本 $\cos t(S)$ 是否优于已知最优解的成本 $\cos t(S^*)$。若优于，则将 S 赋值给已知最优解 S^*，再返回第 2 步。

(4) 将最优邻域解 S 赋值给当前解 $S_{current}$，并将 S 记录至禁忌表。

(5) 若迭代次数达到上界，算法结束，否则返回至第 2 步。

禁忌搜索的伪代码如表 3.4 所示

表 3.4 禁忌搜索的伪代码

Algorithm 3.2: Tabu Search

1. 生成初始解 $S_{current}$，生成列表 Tabu = [$S_{current}$]
2. **While** *the stopping criterion is not met* **do**
3. **For** $S \in N(S_{current})$
4. **If** S in Tabu **then**
5. 检验破禁准则来决定是计算 S 的成本，还是返回 For 循环
6. **If** $\cos t(S) < \cos t(S^*)$ **then**
7. $S^* := S$
8. $S_{current} := S$,**where** $\cos t(S)$ is minimal
9. Tabu add $S_{current}$

3.3 变邻域搜索

3.3.1 变邻域搜索的要素

Mladenovi and Hansen 所提出的变邻域搜索(Variable Neighborhood Search, VNS)基于爬山法的邻域搜索，并通过多类邻域动作来尽可能跳出局部最优解[39]。

综上，变邻域搜索的基本要素为多类邻域动作，作用是交替使用以跳出局部最优解。

3.3.2 变邻域搜索的流程

表 3.5 符号说明

符号	说明
S	问题的解
$S_{current}$	当前解
S^*	已知最优解
$\cos t(S)$	解 S 对应的成本值
$N_i(S_{current})$	当前解的第 i 类邻域空间
$LocalSearch(SolutionSet)$	返回 $SolutionSet$ 中成本最优的解

变邻域搜索的流程为：

(1) 生成初始解作为当前解 $S_{current}$。

(2) 搜索当前解 $S_{current}$ 的邻域最优解 S，比较其成本 $\cos t(S)$ 与当前解的成本 $\cos t(S_{current})$。如果优于当前解的成本，则将 S 赋值给 $S_{current}$，重新运行第 2 步，否则运行第 3 步。

(3) 选择新的邻域动作并得到新的邻域空间，返回第 2 步。若已无新的邻域动作可选，则算法结束，得到的当前解 $S_{current}$ 即已知最优解 S^*。

变邻域搜索的伪代码如表 3.6 所示：

表 3.6 变邻域搜索的伪代码

Algorithm 3.3: Variable Neighborhood Search

1. 生成初始解 $S_{current}$
2. **While** $i <= k$ **do**
3. $S := LocalSearch(N_i(S_{current}))$

Algorithm 3.3: Variable Neighborhood Search

4. **If** $\cos t(S) < \cos t(S_{current})$ **then**
5. $S_{current} := S$
6. $i := 1$
7. **else**
8. $i := i + 1$

第 4 章　MDPVRP 与细邻域禁忌搜索-变邻域搜索

4.1　周期性多车场车辆路径问题

4.1.1　问题的数学模型

本文研究的 MDPVRP 问题以最小化行驶总路程为目标，并考虑车辆载重和车辆用时的限制条件。为了详细地进行说明，本条将先在表 4.1 中介绍各种参数和变量，再介绍 MDPVRP 问题的数学模型。

表 4.1　参数和变量说明

符号	说明
V	所有车场和所有客户的集合
V^{DEP}	所有车场的集合
V^{CST}	所有客户的集合
V_i	表示节点 i
V_o	表示一个车场 o
p	表示一个配送日期组合
l	表示一个配送日期
t	配送日期的上限
k	表示一个车辆
m	车辆数量上限
Q	车辆的载重
D	车辆的用时上限
L_i	客户 i 配送日期组合的集合
q_i	客户 i 的货物需求
d_i	服务客户 i 所需的服务时间
c_{ij}	从 i 至 j 的距离
y_{ip}	0-1 变量。当且只当客户 i 选择配送日期组合 p 时，$y_{ip}=1$
a_{pl}	0-1 变量。当且只当配送日期组合 p 包括日期 l 时，$a_{pl}=1$
x_{ijklo}	0-1 变量。当且只当在日期 l，由车场 o 的车辆 k 访问节点 i 后又立即访问节点 j 时，$x_{ijklo}=1$

MDPVRP 问题的数学模型为：

$$\text{Minimize} \sum_{i\in V}\sum_{j\in V}\sum_{k=1}^{m}\sum_{l=1}^{t}\sum_{o\in V^{DEP}} c_{ij}x_{ijklo} \tag{4.1}$$

$$Subject\quad to: \sum_{p\in L_i} y_{ip} = 1 \qquad V_i \in V^{CST} \tag{4.2}$$

$$\sum_{j\in V}\sum_{k=1}^{m}\sum_{V_o\in V^{DEP}} x_{ijklo} = \sum_{p\in L_i} a_{pl}y_{ip} \quad V_i \in V^{CST}; l = 1...t \tag{4.3}$$

$$\sum_{i\in V} x_{ihklo} = \sum_{j\in V} x_{hjklo} \quad V_h \in V; V_o \in V^{DEP};\ \ k = 1...m; l = 1...t \tag{4.4}$$

$$\sum_{j\in V} x_{ojklo} \le 1 \quad V_o \in V^{DEP}; k = 1...m; l = 1...t \tag{4.5}$$

$$\sum_{i\in V}\sum_{j\in V} q_i x_{ijklo} \le Q \quad V_o \in V^{DEP}; k = 1...m; l = 1...t \tag{4.6}$$

$$\sum_{i\in V}\sum_{j\in V} (c_{ij} + d_i)x_{ijklo} \le D \quad V_o \in V^{DEP}; k = 1...m; l = 1...t \tag{4.7}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijklo} \le |S| - 1 \quad V_o \in V^{DEP}; S \in V^{CST}; |S| >= 2; k = 1...m; l = 1...t; \tag{4.8}$$

$$y_{ip} = \{0,1\} \quad i \in V^{CST}; p \in L_i \tag{4.9}$$

$$a_{pl} = \{0,1\} \quad p \in L_i; l = 1...t \tag{4.10}$$

$$x_{ijklo} = \{0,1\} \quad V_i, V_j \in V; V_o \in V^{DEP}; k = 1...m; l = 1...t \tag{4.11}$$

表达式(4.1)意味着优化目标是最小化行驶路程总和。约束(4.2)意味着必须为每

个客户选择一种配送日期组合。约束(4.3)意味着只有在客户 i 的配送日期组合里包括了日期 l 时，才在日期 l 对客户 i 进行配送。约束(4.4)意味着一辆车为客户 i 完成服务后必须离开客户 i。约束(4.5)意味着一个车场的一辆车一天只能使用一次。约束(4.6)意味着一辆车所服务的客户群的货物需求量不能超过车辆载重。约束(4.7)意味着一辆车所服务客户群的用时不能超过车辆用时上限。约束(4.8)中的 S 是 V^{CST} 的子集，该约束用于消除子回路。

4.1.2 问题的分解

本文解决 MDPVRP 问题的思路是先为每个客户 i 分配配送日期组合，再为其分配车场，最后为每个车场规划配送路径。相较于 Hadjiconstantinou and Baldacci 先为客户群分配车场，再决策配送日期的思路[24]，本文分解问题的思路可以搜索到更多的解，这是因为倘若先为客户群分配车场，则忽略了一个客户在不同的日期可以由不同的车场服务，从而忽略了部分可行解。

综上，本文将 MDPVRP 问题分解为三个部分：

(1) 为每个客户 i 分配配送日期组合，并使用细邻域禁忌搜索优化分配方案。

(2) 确定所有客户的配送日期组合后，为每个客户 i 分配车场，并使用细邻域禁忌搜索优化分配方案。

(3) 确定所有客户分配的车场后，为每个车场规划配送路径，并使用变邻域搜索优化。

4.2 细邻域禁忌搜索-变邻域搜索算法

针对 MDPVRP 问题分解而成的三个部分，本节将按如下顺序分别加以解决，进而解决 MDPVRP 问题。

(1) 针对配送日期组合和车场的分配分别提出两个细邻域搜索准则。

(2) 针对配送日期组合和车场的分配分别确定禁忌搜索的四个要素：邻域动作、禁忌对象、破禁准则和禁忌表长度。

(3) 针对车场配送路径的规划设计变邻域搜索。

(4) 将细邻域搜索准则、禁忌搜索的要素以及变邻域搜索合成，确定细邻域禁忌搜索-变邻域搜索算法的详细流程，以解决 MDPVRP 问题。

4.2.1 细邻域搜索准则的设计

该部分将分别介绍为客户分配配送日期组合、分配车场的细邻域搜索准则。

1. 分配配送日期组合的细邻域搜索准则

表 4.2 参数、变量及符号说明

符号	说明
S^p	配送日期组合分配的一个解
$S^p_{current}$	配送日期组合分配的当前解
$P(S^p)$	解 S^p 的评估函数
$N(S^p_{current})$	当前解 $S^p_{current}$ 的邻域空间
$\cos t(S^p)$	解 S^p 的成本函数
$std(DataSet)$	数据集 $DataSet$ 的标准差
V^{CST}	所有客户的集合
i	表示一个客户
l	表示一个配送日期
t	配送日期的上限
q_i	客户 i 的货物需求
d_i	服务客户 i 所需的服务用时
b_{il}	0-1 变量。当且只当客户 i 于日期 l 被服务时，$b_{il}=1$
Dem_l	日期 l 要配送的客户的需求总和
Dur_l	日期 l 要配送的客户的服务用时总和
$demwgt$	Dem_l 的权重
$durwgt$	Dur_l 的权重
$WorkLoad_l$	Dem_l 与 Dur_l 的加权
$prob$	控制细邻域大小的常数

该部分将提出的细邻域搜索准则基于“需求平摊原则”，即一个好的解 S^p 应具有如下性质：不同日期客户群的需求总和 Dem_l 以及服务用时总和 Dur_l 的加权 $WorkLoad_l$ 应较为均匀，不应出现某一天的加权 $WorkLoad_l$ 特别大或者特别小。为此，下文先定义 Dem_l、Dur_l、$WorkLoad_l$ 以及 $P(S^p)$，再介绍搜索流程。

日期 l 的需求总和 Dem_l 定义如公式(4.12)所示，日期 l 的服务用时总和 Dur_l 定

义如公式(4.13)所示。

$$Dem_l = \sum_{i \in V^{CST}} b_{il} q_i \tag{4.12}$$

$$Dur_l = \sum_{i \in V^{CST}} b_{il} d_i \tag{4.13}$$

日期l的需求总和与服务用时总和的的加权$WorkLoad_l$定义为公式(4.14)所示。

$$WorkLoad_l = demwgt \cdot Dem_l + durwgt \cdot Dur_l \tag{4.14}$$

评估$\{WorkLoad_l, l = 1...t\}$均匀程度的表达式如公式(4.15)所示。

$$std(\{WorkLoad_l, l = 1...t\}) \tag{4.15}$$

解S^p的评估函数$P(S^p)$如公式(4.16)所示。

$$P(S^p) = std(\{WorkLoad_l, l = 1...t\}) \tag{4.16}$$

分配配送日期组合的细邻域搜索流程为：

(1) 生成初始解作为当前解$S^p_{current}$

(2) 计算当前解的评估函数$P(S^p_{current})$

(3) 从当前解$S^p_{current}$的邻域$N(S^p_{current})$中选择一个邻域解S^p并计算S^p的评估函数$P(S^p)$。

(4) 以概率$(P(S^p_{current}) / P(S^p))^{\wedge} prob$决定是否对邻域解$S^p$进行成本运算。如果不进行成本运算，则返回第 3 步；如果进行成本运算，则先计算$\cos t(S^p)$，再返回第 3 步。

(5) 将成本运算得到的具有最小成本的S^p赋值给$S^p_{current}$作为新的当前解。

(6) 判断程序迭代次数是否达到上限。如果未达到上限，则返回第 2 步；如果已达到上限，则算法结束并返回成本运算得到的最小值。

上述流程的伪代码如表 4.3 所示：

表 4.3 分配配送日期组合的细邻域搜索伪代码

1. 生成初始解作为当前解 $S^p_{current}$ 并计算评估函数 $P(S^p_{current})$
2. **While** 迭代数未达临界值 **do**
3. **For** $S^p \in N(S^p_{current})$
4. 计算评估函数 $P(S^p)$
5. 随机生成(0,1)内的随机数 r
6. **If** $r <= (P(S^p_{current}) / P(S^p))\hat{}\ prob$ **then**
7. 计算成本函数 $\cos t(S^p)$
8. **else**
9. **continue**
10. $S^p_{current} := S^p$,**where** $\cos t(S^p)$ is minimal
11. 迭代数+1

可见，如果解 S^p 对应的 $WorkLoad_l$ 在不同日期 l 差异较大，则 $P(S^p)$ 数值会较大，从而 $(P(S^p_{current}) / P(S^p))\hat{}\ prob$ 较小，使得 S^p 不太可能进行成本计算。综上，以上评估函数的定义可以忽略 $WorkLoad_l$ 较不均匀的解 S^p，从而减小邻域空间，加速邻域搜索。

2.分配车场的细邻域搜索准则

表 4.4 参数、变量及符号说明

符号	说明
S^d	车场分配的一个解
$S^d_{current}$	车场分配的当前解
$NM(S^d_{current})$	当前解 $S^d_{current}$ 的邻域动作的集合
$\cos t(S^d)$	解 S^d 的成本函数
$cv(DataSet)$	数据集 $DataSet$ 的变异系数
$avg(DataSet)$	数据集 $DataSet$ 的平均数
V^{CST}	所有客户的集合
V^{DEP}	所有车场的集合
i, j	表示一个客户或一个车场

符号	说明
o	表示一个车场
c_{ij}	从 i 至 j 的距离
$move(i)$	改变 i 所分配的车场的邻域动作
$D(move(i))$	邻域动作 $move(i)$ 的评估函数
dis_i	i 距离各个车场的距离集合
$prob$	控制细邻域大小的常数

该部分将提出的细邻域搜索准则基于“就近分配原则”，即一个好的邻域动作 $move(i)$ 应有如下性质：节点 i 距离各个车场的距离 dis_i 的变异系数 $cv(dis_i)$ 较小，否则节点 i 应就近分配车场而不该被邻域动作所变动。使用变异系数而非标准差来衡量 dis_i 差异度的原因在于：变异系数不受数据量级的影响，从而在 dis_i 的量级较大或较小时都能反应数值的离散程度。为此，下文先定义 dis_i 及领域动作 $move(i)$ 的评估函数 $D(move(i))$，再介绍搜索流程。

节点 i 距离各个车场的距离 dis_i 的定义如公式(4.17)所示。

$$dis_i = \left\{ c_{io}, o \in V^{DEP} \right\} \quad i \in V^{CST} \tag{4.17}$$

领域动作 $move(i)$ 的评估函数 $D(move(i))$ 的定义如公式(4.18)所示。

$$D(move(i)) = cv(dis_i) \tag{4.18}$$

分配车场的细邻域搜索流程为：

(1) 生成初始解作为当前解 $S^d_{current}$

(2) 从当前解 $S^d_{current}$ 的邻域动作集合 $NM(S^d_{current})$ 中选择一个邻域动作 $move(i)$ 并计算评估函数 $D(move(i))$。

(3) 以概率 $(avg(\{D(move(j)), j \in V^{CST}\}) / D(move(i)))\hat{}\ prob$ 决定是否对当前解 $S^d_{current}$ 使用邻域动作 $move(i)$。如果不使用邻域动作 $move(i)$，则返回第 2 步；如果使用，则先对当前解 $S^d_{current}$ 使用邻域动作 $move(i)$ 生成邻域解 S^d，再计算邻域解的成本 $\cos t(S^d)$，最后返回第 2 步。

(4) 将成本运算得到的具有最小成本的 S^d 赋值给 $S^d_{current}$ 作为新的当前解。

(5) 判断程序迭代次数是否达到上限。如果未达到上限，则返回第 2 步；如果

已达到上限，则算法结束并返回成本运算得到的最小值。

上述流程的伪代码如表 4.5 所示：

表 4.5 分配车场的细邻域搜索伪代码

1. 生成初始解 $S^d_{current}$
2. **While** 迭代数未达临界值 **do**
3. **For** $move(i) \in NM(S^d_{current})$
4. 计算评估函数 $D(move(i))$
5. 随机生成(0,1)内的随机数 r
6. **If** $r <= (avg(\{D(move(j)), j \in V^{CST}\}) / D(move(i)))^{\wedge} prob$ **then**
7. 对 $S^d_{current}$ 使用邻域动作 $move(i)$ 得到邻域解 S^d
8. 计算成本函数 $\cos t(S^d)$
9. **else**
10. **continue**
11. $S^d_{current} := S^d$,**where** $\cos t(S^d)$ is minimal
12. 迭代数+1

可见，如果节点 i 距离各个车场的距离 dis_i 的变异系数 $cv(dis_i)$ 较大，则 $D(move(i))$ 数值会较大，从而 $(avg(\{D(move(j)), j \in V^{CST}\}) / D(move(i)))^{\wedge} prob$ 数值较小，使得 $S^d_{current}$ 不太可能使用邻域动作 $move(i)$。综上，以上评估函数的定义可以忽略距离变异系数 $cv(dis_i)$ 较大的节点 i 的邻域动作 $move(i)$，从而减小邻域空间，加速邻域搜索。

4.2.2 禁忌搜索要素设定

该部分将介绍为客户分配配送日期组合、分配车场的禁忌搜索算法的要素。

1. 分配配送日期的禁忌搜索算法的邻域动作、禁忌对象、破禁准则与禁忌表长度

表 4.6 参数、变量及符号说明

符号	说明
V^{CST}	所有客户的集合
i	表示一个客户

符号	说明
$move(i)$	改变 i 所分配的配送日期组合的邻域动作
t	配送日期的上限
$scale$	控制禁忌表长度的常数

邻域动作： $move(i)$ 表示在 L_i 中为客户 i 选择新的配送日期组合。

禁忌对象：如果改变客户 i 的配送日期组合，则将客户 i 作为禁忌对象记录到禁忌表中。

破禁准则：如果客户 i 已在禁忌表中，但改变客户 i 的配送日期组合得到解 S^p 后，解 S^p 的成本优于已知最优解的成本，则允许对客户 i 改变配送日期组合。

禁忌表长度：为了使禁忌表与配送周期的长短和客户群的规模相适应，定义禁忌表长度为 $scale\cdot\sqrt{|V^{CST}|\cdot t}$ ，并由常数 $scale$ 控制禁忌表的具体长度。

2. 分配车场的禁忌搜索算法的邻域动作、禁忌对象、破禁准则与禁忌表长度

表 4.7　参数、变量及符号说明

符号	说明
V^{CST}	所有客户的集合
V^{DEP}	所有车场的集合
i	表示一个客户
o_1, o_2	表示 o_1 和 o_2 两个车场
m	一个车场的车辆数上限
$scale$	控制禁忌表长度的常数

邻域动作：在 V^{DEP} 中为客户 i 选择新的车场。

禁忌对象：如果将客户 i 从车场 o_1 转移至车场 o_2 ，则将集合 $\{o_1, o_2, i\}$ 作为禁忌对象记录到禁忌表中。

破禁准则：如果将客户 i 从车场 o_1 转移至车场 o_2 得到解 S^d 后，解 S^d 的成本优于已知最优解的成本，则允许将客户 i 从车场 o_1 转移至车场 o_2 。

禁忌表长度：定义禁忌表长度为 $scale\cdot\sqrt{|V^{CST}|+|V^{DEP}|\cdot m}$ ，并由常数 $scale$ 控制禁忌表的具体长度。

4.2.3 变邻域搜索的设计

该部分将介绍为一个车场规划配送路径的变邻域搜索。

表 4.8 参数、变量及符号说明

符号	说明
$S_{current}^{c}$	车场配送路径规划的当前解
S^{c}	车场配送路径规划的一个解
$N_1(S_{current}^{c})$	$S_{current}^{c}$ 使用 $2-opt^{*}$ 作为邻域动作生成的邻域空间
$N_2(S_{current}^{c})$	$S_{current}^{c}$ 使用 $2-opt$ 作为邻域动作生成的邻域空间
$N_3(S_{current}^{c})$	$S_{current}^{c}$ 使用 *relocate* 作为邻域动作生成的邻域空间
$\cos t(S^{c})$	解 S^{c} 对应的成本
$LS(N_i(S_{current}^{c}))$	对 $S_{current}^{c}$ 的第 i 个邻域空间进行邻域搜索，返回该邻域空间的最优邻域解

所使用的三类邻域动作 $2-opt^{*}$、$2-opt$ 和 *relocate* 的具体细节为：

(1) $2-opt^{*}$ 是将两辆车的路径进行部分交换。如果两辆车的路径分别为 $[1,2|3,4,5]$ 和 $[6,7|8,9,10]$，那么交换后两辆车的路径分别为 $[1,2|8,9,10]$ 和 $[6,7|3,4,5]$。

(2) $2-opt$ 是将一辆车路径的一部分进行逆序排列。如果一辆车的路径为 $[1|2,3,4|5]$，则逆序排列后的路径为 $[1|4,3,2|5]$。

(3) *relocate* 是在一辆车的路径内改变一个点的位置。如果一辆车的路径为 $[|1|2,3,4|5|]$，则改变点的位置后，路径为 $[2,3,4,1,5]$。

为一个车场规划配送路径的变邻域搜索流程为：

(1) 使用 C-W 节约算法[7]生成初始解作为当前解 $S_{current}^{c}$。

(2) 令 $S^{c}=LS(N_1(S_{current}^{c}))$，如果 $\cos t(S^{c})<\cos t(S_{current}^{c})$，则令 $S_{current}^{c}:=S^{c}$，并再次执行第 2 步。如果 $\cos t(S^{c})>=\cos t(S_{current}^{c})$，则转至第 3 步。

(3) 令 $S^{c}=LS(N_2(S_{current}^{c}))$，如果 $\cos t(S^{c})<\cos t(S_{current}^{c})$，则令 $S_{current}^{c}:=S^{c}$，并转至第 2 步。如果 $\cos t(S^{c})>=\cos t(S_{current}^{c})$，则转至第 4 步。

(4) 令 $S^{c}=LS(N_3(S_{current}^{c}))$，如果 $\cos t(S^{c})<\cos t(S_{current}^{c})$，则令 $S_{current}^{c}:=S^{c}$，并转至第 2 步。如果 $\cos t(S^{c})>=\cos t(S_{current}^{c})$，算法结束并返回整个搜索过程的最小成本。

4.2.4 细邻域禁忌搜索-变邻域搜索的实现流程

表 4.9 参数、变量及符号说明

符号	说明
$S_{current}^{p}$	配送日期组合分配的当前解
S^{p}	配送日期组合分配的一个解
S^{p*}	配送日期组合分配的已知最优解
$P(S^{p})$	解 S^{p} 的评估函数
$S_{current}^{d}$	车场分配的当前解
S^{d}	车场分配的一个解
S^{d*}	车场分配的已知最优解
$S_{current}^{c}$	车场配送路径规划的当前解
S^{c}	车场配送路径规划的一个解
S^{c*}	车场配送路径规划的已知最优解
$N_{p}(S_{current}^{p})$	当前解 $S_{current}^{p}$ 的邻域空间
$NM_{d}(S_{current}^{d})$	当前解 $S_{current}^{d}$ 的邻域动作的集合
$N_{1}(S_{current}^{c})$	当前解 $S_{current}^{c}$ 使用 $2-opt^{*}$ 作为邻域动作生成的邻域空间
$N_{2}(S_{current}^{c})$	当前解 $S_{current}^{c}$ 使用 $2-opt$ 作为邻域动作生成的邻域空间
$N_{3}(S_{current}^{c})$	当前解 $S_{current}^{c}$ 使用 $relocate$ 作为邻域动作生成的邻域空间
$\cos t(Solution)$	解 $Solution$ 对应的成本
$LS(N_{i}(S_{current}^{c}))$	对当前解 $S_{current}$ 的第 i 个邻域空间进行邻域搜索，返回该邻域空间的最优邻域解
$V_{l}(S^{p})$	当配送日期组合分配为 S^{p} 时，在日期 l 要服务的客户的集合
$V_{o}(S^{d})$	当车场分配为 S^{d} 时，车场 o 要服务的客户的集合
V^{DEP}	所有车场的集合
V^{CST}	所有客户的集合
$D(move(i))$	邻域动作 $move(i)$ 的评估函数

部分符号的具体形式为：

$S^p=\left\{i:P_i,i\in V^{CST}\right\}$ 。P_i为日期的集合，如$P_i=\{1,2\}$表示要在星期一和星期二访问客户i。

$S^d=\left\{o:D_o,o\in V^{DEP}\right\}$。$D_o$为客户的集合，如$D_{o1}=\{1,2\}$表示车场$o_1$要服务客户1和客户2，$D_{o2}=\{3,4\}$表示车场$o_2$要服务客户3和客户4。

$S^c=\left\{k:C_k,k=1...m\right\}$。$C_k$为客户的顺序，如$C_k=[1\to 2\to 3]$表示车辆$k$从车场出发先服务客户1，再服务客户2，最后服务客户3并返回车场。

令$overdem(Solution)$和$overdur(Solution)$分别表示解$Solution$因为超出车辆载重限制和用时限制而产生的罚成本，则

$$\cos t(S^c)=\sum_{i\in V}\sum_{j\in V}\sum_{k=1}^{m}c_{ij}x_{ijklo}+overdem(S^c)+overdur(S^c)$$

$$\cos t(S^d)=\sum_{i\in V}\sum_{j\in V}\sum_{k=1}^{m}\sum_{o\in V^{DEP}}c_{ij}x_{ijklo}+overdem(S^d)+overdur(S^d)$$

$$\cos t(S^p)=\sum_{i\in V}\sum_{j\in V}\sum_{k=1}^{m}\sum_{l=1}^{t}\sum_{o\in V^{DEP}}c_{ij}x_{ijklo}+overdem(S^p)+overdur(S^p)$$

$$N_p(S^p_{current})=\{\{i:P_i^{'},i\in V^{CST}\},\{i:P_i^{''},i\in V^{CST}\}...\}$$

$$NM_d(S^d_{current})=\{move(i),move(j)...\}$$

细邻域禁忌搜索-变邻域搜索的详细流程为：

模块1：

(1) 为每个客户i随机分配配送日期组合，将分配方法作为当前解$S^p_{current}$并计算评估函数$P(S^p_{current})$。

(2) 从当前解$S^p_{current}$的邻域$N_p(S^p_{current})$中选择一个邻域解S^p并计算S^p的评估函数$P(S^p)$。

(3) 以概率$(P(S^p_{current})/P(S^p))\hat{}\ prob$决定是否对邻域解$S^p$进行成本运算。如果不进行成本运算，则返回第2步；如果进行成本运算，则运行模块2获得邻域解S^p的成本$\cos t(S^p)$，返回第2步。

(4) 取成本运算得到的具有最小成本的S^p，判断得到S^p所改变的客户i是否已在禁忌表中。若不在，则令$S^p_{current}=S^p$。若在禁忌表中，则检验破禁准则。若允许破禁，则令$S^p_{current}=S^p$，若不允许破禁，则取成本次小的解S^p，重复之前的操作，

以此类推，直至令 $S^p_{current} = S^p$ 为止。将 S^p 所改变的客户 i 记录到禁忌表中，并对禁忌表中任期达到上限的禁忌对象解禁。

(5) 判断程序迭代次数是否达到上限。如果未达到上限，则返回第 2 步；如果已到上限，则算法结束并返回整个搜索过程得到的最小成本 $\cos t(S^{p*})$ 作为 MDPVRP 问题的最优解。

模块 2：

(1) 根据 S^p，求得不同日期要服务的客户集合 $\{V_l(S^p), l = 1...t\}$。取日期 l 的客户集合 $V_l(S^p)$。

(2) 对于日期 l 的客户集合 $V_l(S^p)$，把每个客户 i 分配至距离最近的车场，将分配方法作为当前解 $S^d_{current}$。

(3) 从当前解 $S^d_{current}$ 的邻域动作集合 $NM(S^d_{current})$ 中选择一个邻域动作 $move(i)$ 并计算评估函数 $D(move(i))$。

(4) 以概率 $(avg(\{D(move(j)), j \in V_l(S^p)\}) / D(move(i)))^{\wedge} prob$ 决定是否对当前解 $S^d_{current}$ 使用邻域动作 $move(i)$。如果不使用邻域动作，则返回第 3 步；如果使用，则先对当前解 $S^d_{current}$ 使用邻域动作 $move(i)$ 生成邻域解 S^d，再运行模块 3 获得邻域解 S^d 的成本 $\cos t(S^d)$，返回第 3 步。

(5) 取成本运算得到的具有最小成本的 S^d，判断得到 S^d 所涉及的邻域动作 $move(i)$ 是否已在禁忌表中。若不在，则令 $S^d_{current} = S^d$。若在禁忌表中，则检验破禁准则。若允许破禁，则令 $S^d_{current} = S^d$，若不允许破禁，则取成本次小的解 S^d，重复之前的操作，以此类推，直至令 $S^d_{current} = S^d$ 为止。将 S^d 所涉及的邻域动作 $move(i)$ 记录到禁忌表中，并对禁忌表中任期达到上限的禁忌对象解禁。

(6) 判断程序迭代次数是否达到上限。如果未达到上限，则返回第 3 步；如果已到上限，输出日期 l 的最小成本 $\cos t_l(S^{d*})$，并返回至第 1 步，对下一个日期 l 则进行类似操作。

(7) 计算 $\cos t(S^p) = \sum_{l=1}^{t} \cos t_l(S^{d*})$，退出模块 2 并输出 $\cos t(S^p)$。

模块 3：

(1) 根据 S^d，求得不同车场要服务的客户集合 $\{V_o(S^d), o \in V^{DEP}\}$。取车场 o 的客户集合 $V_o(S^d)$。

(2) 对于车场o的客户集合$V_o(S^d)$，使用 C-W 节约算法生成当前解$S^c_{current}$。

(3) 令$S^c = LS(N_1(S^c_{current}))$。

如果$\cos t(S^c) < \cos t(S^c_{current})$，则令$S^c_{current} := S^c$，再次运行第 3 步。如果$\cos t(S^c) > \cos t(S^c_{current})$，则令$S^c = LS(N_2(S^c_{current}))$。

如果$\cos t(S^c) < \cos t(S^c_{current})$，则令$S^c_{current} := S^c$，再次运行第 3 步。如果$\cos t(S^c) > \cos t(S^c_{current})$，则令$S^c = LS(N_3(S^c_{current}))$。

如果$\cos t(S^c) < \cos t(S^c_{current})$，则令$S^c_{current} := S^c$，再次运行第 3 步。如果$\cos t(S^c) > \cos t(S^c_{current})$，结束第 3 步并输出车场$o$的最优值$\cos t_o(S^{c*})$，并返回至第 1 步，对下一个车场$o$则进行类似操作。

(4) 计算$\cos t(S^d) = \sum_{o \in V^{DEP}} \cos t_o(S^{c*})$，退出模块 3 并输出$\cos t(S^d)$。

第 5 章　标准数据集的优化

本章将细邻域禁忌搜索-变邻域搜索算法(GTS-VNS)应用于 MDPVRP 标准数据集的优化，以体现新算法的潜力。

5.1　算法参数的设置

GTS-VNS 算法的参数组为$(demwgt, durwgt, prob, scale)$，为了获得最适合的参数，将针对标准数据集中的$dataset01$进行数值实验调参。每组参数都运行 10 次程序，并取最优值$\cos t^*$、平均最优值$avg\cos t$、平均收敛用时$avgtime(\min)$三个指标衡量参数组的优劣。由于评估函数$P(S^p)$与$demwgt$、$durwgt$的比值有关，而非与单个$demwgt$或$durwgt$的值相关，所以实际上只需固定$demwgt$的值并调节参数组$(durwgt, prob, scale)$即可。

0.$(demwgt, durwgt, prob, scale)$初始化为$(1,1,0,1)$

1.调节 $prob$。固定 $demwgt=1, durwgt=1, scale=1$，令 $prob$ 在集合$\{0,20,40,60,80,100\}$中取值，所得的优化结果如表 5.1 所示：

表 5.1 参数 $prob$ 的调节

$demwgt=1, durwgt=1, scale=1$			
$prob$	$\cos t^*$	$avg\cos t$	$avgtime(\min)$
0	2057.91	2134.34	1.70
20	**2045.60**	**2086.12**	1.85
40	**2046.12**	**2100.88**	**1.27**
60	2055.57	2116.18	1.81
80	2050.70	2113.49	1.77
100	2046.65	2114.26	**1.29**

可见，当$prob=40$时，算法在最优值、平均最优值和平均收敛用时方面皆排名前 2，是三类指标都排名前 2 的唯一数值，故选择$prob=40$。

2.调节 $durwgt$。在选择$prob=40$后，固定$demwgt=1, prob=40, scale=1$，令 $durwgt$ 在集合$\{0,0.5,1,1.5,2\}$中取值，所得的优化结果如表 5.2 所示：

表 5.2 参数 *durwgt* 的调节

$demwgt=1, prob=40, scale=1$			
durwgt	$\cos t^*$	$avg\cos t$	*avgtime*(min)
0	2046.98	**2081.54**	1.99
0.5	**2044.62**	2119.26	**1.20**
1	2053.78	2111.73	**1.57**
1.5	2047.59	**2089.76**	1.74
2	**2044.78**	2097.92	1.96

可见，当 $durwgt=0.5$ 时，算法在最优值和平均收敛用时方面皆排名前 2，是有两类指标都排名前 2 的唯一数值，故选择 $durwgt=0.5$。

3. 调节 *scale*。在选择 $prob=40$ 以及 $durwgt=0.5$ 后，固定 $demwgt=1$，$prob=40$，$durwgt=0.5$，令 *scale* 在集合 {0.5,1,1.5,2,2.5} 中取值，所得的优化结果如表 5.3 所示：

表 5.3 参数 *scale* 的调节

$demwgt=1, durwgt=0.5, prob=40$			
scale	$\cos t^*$	$avg\cos t$	*avgtime*(min)
0.5	**2046.65**	2120.95	2.12
1	**2044.62**	**2119.26**	1.20
1.5	2056.81	2127.66	**0.83**
2	**2055.90**	**2112.53**	**0.94**
2.5	2060.09	**2112.03**	**0.69**

可见，当 $scale=2$ 时，算法在最优值、平均最优值和平均收敛用时方面皆排名前 3，是三类指标都排名前 3 的唯一数值，故选择 $scale=2$。

综上，GTS-VNS 算法的参数组为 $(demwgt=1, durwgt=0.5, prob=40, scale=2)$

5.2 优化结果与分析

MDPVRP 问题的标准数据集由 10 组数据集构成，分别记为 $pr01, pr02...pr10$。本节将给出 GTS-VNS 算法优化各个数据集得出的最优值 $\cos t^*$、平均最优值 $avg\cos t$、平均收敛用时 $avgtime$(min)，并与学术界已知的最优值(Best Known Solution, BKS)作比较。为了保证求解的客观性，对每组数据集都将运行 10 次求解程序，所使用的求解语言和硬件设备为：Python3 编程，Intel i5 处理器，8GB 运行内存。除此之外，本节还将展示 GUROBI 的优化结果以及 Michel Gendreau et al 提出的 HGSADC 算法[17]的优化结果。

GTS-VNS 算法的优化结果如表 5.4 所示：

表 5.4 GTS-VNS 算法的优化结果

dataset	*n*	*m*	*d*	*t*	*BKS*	$\cos t^*$	$avg\cos t$	*avgtime*(min)	*gap to BKS* (%)
*pr*01	**48**	**1**	**4**	**4**	**2019.07**	**2044.62**	2112.53	0.94	**1.27**
*pr*02	96	1	4	4	3547.45	-	-	-	-
*pr*03	144	2	4	4	4480.87	4786.98	4954.98	19.77	6.83
*pr*04	192	2	4	4	5134.17	-	-	-	-
*pr*05	240	3	4	4	5570.45	-	-	-	-
*pr*06	288	3	4	4	6524.92	-	-	-	-
*pr*07	**72**	**1**	**6**	**6**	**4502.02**	**4725.99**	4772.98	6.94	**4.97**
*pr*08	144	1	6	6	6023.98	-	-	-	-
*pr*09	216	2	6	6	8257.80	9009.61	9135.21	25.59	9.10
*pr*10	388	3	6	6	9818.42	-	-	-	-

注：“-”表示 GTS-VNS 算法收敛时未找到可行解

n、m、d、t 分别表示数据集包含的客户数、每个车场的车辆数、车场数、配送日期的天数。可见，GTS-VNS 算法优化数据集 $pr01$ 与 $pr07$ 所得最优解与学术界已知最优解 BKS 的差距保持在 5%以内；优化数据集 $pr03$ 与 $pr09$ 所得最优解与

BKS 的差距保持在 10%以内；优化其余数据集时都在算法收敛时未找到可行解。

精确算法方面，本文基于 MDPVRP 的数学模型，并使用求解速度最快的优化求解器 GUROBI 进行求解，GUROBI 求解器所使用的精确算法包括分支剪枝法、单纯形法以及一些辅助性的分支策略。以规模最小的数据集 *pr*01 为例，GUROBI 求解器的求解结果如图 5.1 所示：

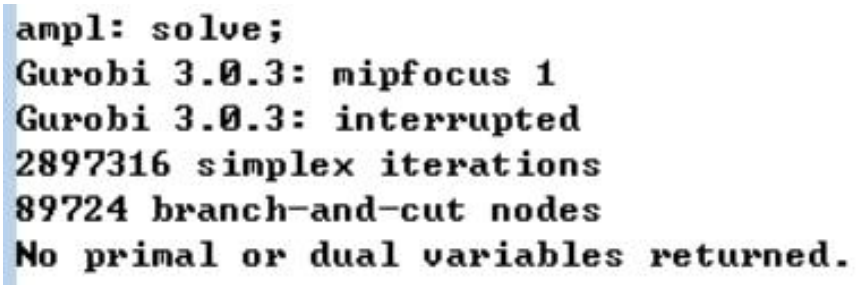

图 5.1 GUROBI 求解结果

可见，即使是规模最小的数据集 *pr*01，GUROBI 也未能返回一个可行解，也就是说，分支剪枝法和单纯形法之类的精确算法无法求解客户节点较多、车场较多、配送周期较长的 MDPVRP，故适用于 MDPVRP 的算法应为启发式或元启发式算法。

Michel Gendreau et al 提出的 HGSADC 算法[17]一度成为 MDPVRP 的最优求解方法，现将 GTS-VNS 算法与 HGSADC 算法的优化结果进行对比，对比数据如表 5.5 所示：

表 5.5 GTS-VNS 算法与 HGSADC 算法对比

dataset	*BKS*	GTS-VNS[a]			HGSADC[b]		
		*c**	*at*(min)	*gap*(%)	*c**	*at*(min)	*gap*(%)
*pr*01	2019.07	2044.62	0.94	1.27	2019.07	0.35	0.00
*pr*03	4480.87	4786.98	19.77	6.83	4480.87	7.72	0.00
*pr*07	4502.02	4725.99	6.94	4.97	4502.02	2.18	0.00
*pr*09	8257.80	9009.61	25.59	9.10	8271.66	27.79	0.17

[a]Python3 编程，Intel i5 处理器，2.50GHz

[b]C++编程，AMD Opteron 250 处理器，2.40GHz

其中，c^*表示运行 10 次程序后所得到的最优值，$at(\min)$ 表示运行 10 次程序所得的平均收敛用时，$gap(\%)$表示算法所得最优值与学术界已知最优值的差距。由于所用编程语言以及处理器不同，GTS-VNS 与 HGSADC 算法的收敛时间无法做直接的比较。但从求解结果来看，HGSADC 算法与已知最优值几乎没有差距，而 GTS-VNS 算法与已知最优值的差距在 1%至 10%之间，且只在较小规模的 *pr*01 与 *pr*07 数据集才有 5%以内的差距，在一般规模和较大规模的 *pr*03 与 *pr*09 数据集有着 10%以内的差距。更值得注意的是，HGSADC 算法在 10 个数据集上都取得了十分优异的结果，而 GTS-VNS 算法只在 4 个数据集上取得了结果，在其余 6 个数据集上都未找到可行解。

通过展示 GTS-VNS 算法、GUROBI 和 HGSADC 算法在标准数据集上的优化结果，可以清楚地看到以下三点：

(1) 使用单纯形法与分支剪枝法的 GUROBI 无法有效求解 MDPVRP，甚至无法找到可行解，必须混合启发式或元启发式算法。

(2) 相比于 Michel Gendreau et al 提出的 HGSADC 算法[17]，目前的 GTS-VNS 算法无论是在求解质量与普适性方面都相形见绌。在求解质量方面，若问题规模较小，HGSADC 算法与已知最优解没有差距，而 GTS-VNS 算法仍会有 5%以内的差距；若问题规模一般或较大，HGSADC 算法与已知最优解的差距近似为 0，而 GTS-VNS 算法会有 10%以内的差距。在普适性方面，HGSADC 算法在全部 10 个的数据集上都能取得结果，而 GTS-VNS 算法只在 4 个数据集上取得了结果。

(3) 在较小规模的数据集中，GTS-VNS 算法可以在 5%以内的差距进行优化求解，相比于无法给出可行解的精确算法，GTS-VNS 算法已属于一种可行的算法。

综上，目前 GTS-VNS 算法的优点是：对于规模较小的 MDPVRP 问题有相对不错的求解效果。目前算法的缺点是：对规模一般或较大的 MDPVRP 问题的求解效果不出色，且普适性不强，可能会因为数据的变动而找不到可行解。

总结与展望

本文针对 MDPVRP 问题，结合细邻域搜索、禁忌搜索和变邻域搜索 3 种求解技术，并基于“需求平摊原则”和“就近分配原则”设计了独特的细邻域搜索准则，从而提出了细邻域禁忌搜索-变邻域搜索算法(GTS-VNS)。GTS-VNS 算法优化标准数据集的结果显示：该算法对于较小规模的 MDPVRP 问题有相对不错的效果，但求解较大规模的问题效果一般，并且普适性不强，可能因为数值的变动而找不到可行解。综上，目前的 GTS-VNS 算法只能说是一种较有潜力的算法，需要加以改进以克服大规模问题求解效果一般、普适性不强的缺陷。

GTS-VNS 算法日后的研究方向主要有如下三点：

(1)使用 Java 或 C++等更快速、更普及的编程语言。由于本文使用的编程语言为 Python3，与学术界普遍使用的 C++相比运行速度过慢，一方面使得算法收敛时间无法与学术界知名算法的收敛时间比较，另一方面也使算法后续开发的速度过慢。

(2)增强算法的普适性。在标准数据集的优化过程中，有 6 个数据集都在算法收敛时未找到可行解，可见目前的 GTS-VNS 极易受数值的影响。下一步应考虑适当放宽“需求平摊原则”和“就近分配原则”的约束，从而保留足够多的可行解，避免算法收敛却找不到可行解的弊端。

(3)优化大规模问题的求解效果。在标准数据集的优化过程中，所有大规模问题的优化结果都不佳，这对于当前的商业要求而言是个巨大的缺点。下一步应专门分析大规模问题的数据结构，并针对其独特的数据结构修改细邻域搜索准则，以优化求解效果。

参考文献

[1] Dantzig G and Ramser J. The trunk dispatching problem[J]. Management Science, 1959, 6(1): 80-91.

[2] E. J. Beltrami and L. D. Bodin. Networks and vehicle routing for municipal waste Collection[J]. Networks, 1974, (4): 65-94.

[3] F. Blakeley, B. Bozkaya, B. Cao, W. Hall, and J. Knolmajer. Optimizing periodic maintenance operations for schindler elevator corporation[J]. Interfaces, 2003, 33(1):67-79.

[4] J. Alegre, M. Laguna, and J. Pacheco. Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts[J]. European Journal of Operational Research, 2007, 179(3): 736-746 .

[5] Balinski M, Quandt R. On an integer program for a delivery problem[J]. Oper Res, 1964, (12): 300-304.

[6] Augerat P, Belenguer JM, Benavent E, Corberán A, Naddef D and Rinaldi G., Computational results with a branch and cut code for the capacitated vehicle routing problem[J/OL].https://www.researchgate.net/publication/239062886_Computational_results_with_a_branch_and_cut_code_for_the_capacitated_vehicle_routing_problem, 1995.

[7] Clarke G., Wright J., Scheduling of vehicles from a central depot to a number of delivery points[J]. Operations Research, 1964, (12): 568–581.

[8] Gillett B., Miller L., A heuristic algorithm for the vehicle dispatch problem[J].Operation Research, 1974, (22):340 – 349.

[9] Glover F., Tabu search. Part I[J]. ORSA Journal on Computing, 1989, (1): 190–206.

[10] Kirkaptrick S., Gelatt C.D., Vecchi M.P. et al. Optimization by simulated annealing[J]. Science, 1983, 220(4598): 671 – 680.

[11] Holland J., Adaptation in Natural and Artificial Systems[N]. University of Michigan Press, Ann Arbor, MI, 1975.

[12] J. Renaud, G. Laporte, and F. F. Boctor. A tabu search heuristic for the multi-depot vehicle routing problem[J]. Comput. Oper. Res., 1996, 23(3): 229 – 235 .

[13] J. F. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems[J]. Networks, 1997, (30):105 – 119.

[14] A. Lim and W. Zhu. A fast and effective insertion algorithm for multi-depot vehicle routing problem with fixed distribution of vehicles and a new simulated annealing approach[A]. In IEA/AIE[C], 2006: 282 – 291.

[15] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems[J]. Computers & Operations Research, 2007, 34(8):2403 – 2435.

[16] B. Ombuki-Berman and T. Hanshar. Using genetic algorithms for multi-depot vehicle routing[A]. In F. B. Pereira and J. Tavares, editors, Bio-inspired Algorithms for the Vehicle Routing Problem[C], Springer, 2009: 77 – 99.

[17] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, Nadia Lahrichi and Walter Rei. A Hybrid Genetic Algorithm for Multi-Depot and Periodic Vehicle Routing Problems[J]. Operations Research, 2012, 60(3): 611-624.

[18] I. M. Chao, B. L. Golden, and E. Wasil. An improved heuristic for the period vehicle routing problem[J]. Networks, 1995, 26(1):25 – 44.

[19] V. C. Hemmelmayr, K. F. Doerner, and R. F. Hartl. A variable neighborhood search heuristic for periodic routing problems[J]. European Journal of Operational Research, 2009, 195(3):791 – 802.

[20] A. C. Matos and R. C. Oliveira. An experimental study of the ant colony system for the period vehicle routing problem[A]. In Ant Colony, Optimization and Swarm Intelligence,Lecture Notes in Computer Science[C], Berlin / Heidelberg: Springer, 2004: 1 – 29.

[21]P. Parthanadee and R. Logendran. Periodic product distribution from multi-depots under limited supplies[J]. IIE Transactions, 2006, 38(11):1009-1026.

[22] Nadia Lahrichi, Teodor Gabriel Crainic, Michel Gendreau, et al. An integrative cooperative search framework for multi-decision-attribute combinatorial optimization: Application to the MDPVRP[J]. European Journal of Operational Research, 2015, 246(2): 400-412.

[23] E. Hadjiconstantinou and R. Baldacci. A multi-depot period vehicle routing problem arising in the utilities sector[J]. Journal of the Operational Research Society,

1998, 49(12): 1239–1248.

[24] W. T. Yang and L. C. Chu. A heuristic algorithm for the multi-depot periodic vehicle routing problem[J]. Journal of Information & Optimization Sciences, 2002, (2):359–367 .

[25] Kang K.H., Y. H. Lee, and B. K. Lee. An Exact Algorithm for Multi Depot and Multi Period Vehicle Scheduling Problem[A]. Gervasi O. et al. Computational Science and Its Applications[C], Berlin/ Heidelberg: Springer, 2005: 350-359.

[26] 柏明国，李金书，韩梅. CVRP 问题的一种启发式算法[J]. 山东科技大学学报（自然科学版），2003，22(4)：52-54.

[27] 汪祖柱，钱家兴，方宏兵，钱付兰. 车辆路径问题的混合优化算法[J]. 运筹与管理，2004，(6)：48-52.

[28] 张春苗，赵燕伟，冷龙龙. 车辆路径问题的水波算法[J]. 数值计算与计算机应用，2018，39(3)：231-242.

[29] 彭碧涛，周永务. 基于禁忌搜索的三维装载车辆路径问题研究[J]. 计算机工程，2011，(11)：190-191.

[30] 戴树贵，陈文兰，潘荫荣，胡幼华. 多配送中心车辆路径安排问题混合蚁群算法[J].四川大学学报（工程科学版），2008，(6)：154-158.

[31] 党立伟，孙小明. 多车场车辆路径问题及混合遗传算法[J]. 科学技术与工程，2012，(8)：1816-1820.

[32] 戚远航，蔡延光，黄何列，蔡颢. 泰森多边形的离散蝙蝠算法求解多车场车辆路径问题[J].控制理论与应用，2018，35(8)：1142-1150.

[33] 曾正洋，许维胜，徐志宇，刘竹馨. 应急物流中的累计时间式多车场车辆路径问题[J]. 控制与决策，2014，29(12)：2183-2188.

[34] 薄非凡，魏法杰. 城市垃圾清运中的周期多车场车辆路径问题[J]. 交通标准化，2009，(19)，104-107.

[35] 齐桐萱. 区域废旧家用电器回收网络规划研究[D]. 大连：大连海事大学，2014.

[36] 刘沙. 周期性回收策略下的医疗废物回收网络优化问题研究[D]. 成都：西南交通大学，2016.

[37] 蔡婉君，王晨宇，于滨，杨忠振，姚宝珍. 改进蚁群算法优化周期性车辆路径问题[J]. 运筹与管理，2014，(5)：70-77.

[38] P. Toth and D. Vigo. The granular tabu search and its application to the vehicle-routing problem[J]. INFORMS Journal on Computing, 2003, 15(4): 333–346.
[39] Hansen P., Mladenović N. An Introduction to Variable Neighborhood Search[M]. Voß S., Martello S., Osman I.H., Roucairol C. Boston: Springer, 1999: 433-458.

致谢

光阴似箭，本科四年的时间一瞬即逝，回故这四年的学习与生活，颇有一番感慨。本科生活即将结束，本科毕业论文成为我最后要上交的答卷，也以此献给一直以来帮助自己的良师益友。

首先，我要感谢我的毕设导师--汪挺松老师。汪老师的课题本身就兼具实用性与技术性，颇符合我的专业与兴趣。且汪老师耐心细致，在许多关键的方面让我受益匪浅，是我能成功完成本科论文的重要因素。

其次，我要感谢四年间为我授课的所有老师们。感谢你们的奉献与付出，让我能领略文明与智慧的奇妙魅力，也感谢你们在我学习和生活上的指导与帮助。

最后，我要感谢我的父母，你们的无私奉献与默默付出为我创造了良好的学习与生活环境，有你们的支持与关爱，我将在未来的道路变得越来越好，同时，为了你们，我也必须要在未来的道路变得越来越好。