武汉理工大学毕业设计（论文）

利用智能混合算法解决大规模路径优化问题

学院（系）： 物流工程学院

专业班级： 物流管理 1001班

学生姓名： 董昊

指导教师： 郑澜波

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名：

年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于1、保密囗，在 年解密后适用本授权书；

2、不保密囗 。

（请在以上相应方框内打“√”）

作者签名： 年 月 日

导师签名： 年 月 日

# 摘 要

车辆路径问题(Vehicle Routing Problem,VRP)作为组合优化问题中的经典问题，一直是该领域研究的热点。同时该问题作为时下快速发展的物流行业的基础，与其有关的研究也变得愈发重要。此外，该问题有着极广的应用范围，比如物流领域中物品的调配与运输，互联网领域中路由寻址的优化，计算机领域数据的分布式存储结构的设计等等。任何涉及到网络的领域都会与车辆路径问题发生关系。因此，对车辆路径问题研究的进步不仅可以促进与该问题有关的理论发展，更能够对现实中各个领域的进步起到重大的推动作用。

本文以车辆路径系统为研究对象，首先对国内外有关VRP问题的研究进行了综述。接着对一些现代启发式算法进行了介绍，并对实验所使用的C++程序的数据结构进行了说明。然后给出了VRP问题的基本描述及其数学模型，并使用智能混合算法对模型进行求解。在此基础上对混合算法与模拟退火法进行了对比。通过利用数值分析，包括均值，方差等对算法产生的解的质量与生成解的时间进行综合比较，从而得出结论。研究结果表明，单纯的使用一种算法会较快的陷入局部最优值。禁忌搜索可以显著提升搜索效率，当两种算法混合使用时，并给予适当的引导，会在合理的时间内生成较好的解。

关键词：路径优化问题；局部搜索；模拟退火法；禁忌搜索；混合算法

# Abstract

The Vehicle Routing Problem, which has been a crucial research area in combinatorial optimization, is also a typical problem in this field. And as the basic of logistic which is progressing rapidly nowadays, the VRP and researches related to it is becoming increasingly magnificent. In addition, the application of this problem is extremely extensive. For instance, the allocation of goods in logistics; the addressing of router; the design of distributed databases storage structures in computer science and so on. The VRP can be applied in any field that is concerned with network. Because all above, researches to VRP would not only promote the evolution of the theory but also advance the progressing of many fields in reality.

This paper mainly focus on VRP. At first the advances and new challenges in VRP are presented. Then this paper introduces several metaheuristic algorithms of solving it and the data structure of the C++ program which the algorithm uses. In the third part the paper presents the description of VRP and its mathematic model. After that the intelligent hybrid algorithm is proposed to solve the problem. The experiment is implemented based on intelligent hybrid algorithm and the result was compared to the result generalized by Simulated Annealing algorithm. Through value analysis includes average and deviation, the quality of the solutions generalized by different algorithms and the time of generalizing these solutions were compared. The result shows that using one algorithm along would cause local minimum easily while using a hybrid of two algorithms with proper guidance would not only lead to good result but also improve the search efficiency greatly.

**Key words**：Vehicle Routing Problem, local search, simulated annealing, tabu search, hybrid algorithm

**目 录**

[摘 要 I](#_Toc388700212)

[Abstract II](#_Toc388700213)

[第1章 绪论 1](#_Toc388700214)

[1.1 论文研究背景与问题提出 1](#_Toc388700215)

[1.2 论文研究的目的与意义 3](#_Toc388700216)

[1.2.1 研究目标与内容 3](#_Toc388700217)

[1.2.2 研究意义 4](#_Toc388700218)

[1.3 论文创新点 4](#_Toc388700219)

[第2章 车辆路径问题研究综述 5](#_Toc388700220)

[2.1 国外路径优化问题研究现状 5](#_Toc388700221)

[2.2 国内路径优化问题研究现状 8](#_Toc388700222)

[第3章 算法及数据结构 10](#_Toc388700223)

[3.1 现代启发式算法 10](#_Toc388700224)

[3.1.1模拟退火法 10](#_Toc388700225)

[3.5.2 禁忌搜索 11](#_Toc388700226)

[3.5.3 引导式邻域搜索 14](#_Toc388700227)

[3.2 数据结构 14](#_Toc388700228)

[3.2.1 数据类型 14](#_Toc388700229)

[3.2.2 解的存储方式 15](#_Toc388700230)

[3.3.3 哈希表示 16](#_Toc388700231)

[3.2.4 各模块之间的关系 17](#_Toc388700232)

[3.3 本章小结 17](#_Toc388700233)

[第4章 带容量约束的车辆路径问题及算法 19](#_Toc388700234)

[4.1 问题提出 19](#_Toc388700235)

[4.2 数学模型构建 20](#_Toc388700236)

[4.2.1 问题假设 20](#_Toc388700237)

[4.2.2 参数定义 20](#_Toc388700238)

[4.2.3 模型建立 20](#_Toc388700239)

[4.3 CVRP的智能混合算法 21](#_Toc388700240)

[4.3.1 初始解的构建 22](#_Toc388700241)

[4.3.2 局部搜索 25](#_Toc388700242)

[4.3.3 退火机制 26](#_Toc388700243)

[4.4.4 集中搜索过程 26](#_Toc388700244)

[4.4 实验和结果 27](#_Toc388700245)

[4.1.1 测试问题 27](#_Toc388700246)

[4.1.2 参数设置 27](#_Toc388700247)

[4.1.3 实验结果 29](#_Toc388700248)

[4.5 本章小结 33](#_Toc388700249)

[第5章 总结与展望 35](#_Toc388700250)

[参考文献（黑体小二，居中） 36](#_Toc388700251)

[附录 37](#_Toc388700252)

[致 谢（黑体小二，居中） 51](#_Toc388700253)

# 第1章 绪论

## 1.1 论文研究背景与问题提出

车辆路径问题是一类组合优化问题，被证明为NP完全问题。该问题最基本形式可描述如下：设有一配送中心，共有M 辆货车，车辆容量为C，有N位顾客，每位顾客有其需求量D。车辆从配送中心出发对客户进行配送服务最后返回场站，要求所有顾客都被配送，每位顾客一次配送完成，且不能违反车辆容量的限制，目的是使所有车辆路线的总距离最小。基于基本车辆路径的框架，研究满足生成运营以及运作需要的各种车辆调配问题，并构建具有高质量以及高鲁棒性的解。

然而，由于VRP问题涉及的因素很多，目前的解决方法只能对特定的问题运用不同的算法求解。经过国内外专家、学者的归纳总结，求解VRP问题时，可以将问题分类为几个简单的组合优化的基本原型，如最短路径问题、旅行商问题(TSP)、中国邮递员问题等，再用与之对应的理论和方法进行求解，最终得到模型最优解或较优解。当前在求解VRP问题时可以将所使用的算法大致分为两类，一类是精确算法，另一类是启发式算法。精确算法主要有分支定界法、割平面法、线性规划法、动态规划法等，其中的主要思想是根据问题抽象出数学模型，再利用纯粹的数学方法进行求解。传统启发式算法主要有构造法、节约法、最邻近法、扫描法、插入法、两阶段法等，其主要思想是通过对智能的模拟，譬如人类的记忆，蚂蚁的寻食等寻找出近似最优解。智能算法主要有遗传算法(Genetic Algorithm, GA)、禁忌搜索算法(Tabu Search, TS)、模拟退火算法(Simulated Annealing, SA)、蚁群算法(AntColony Algorithm, ACA)等。[1]图1.1列出了各种启发式算法及其分类。

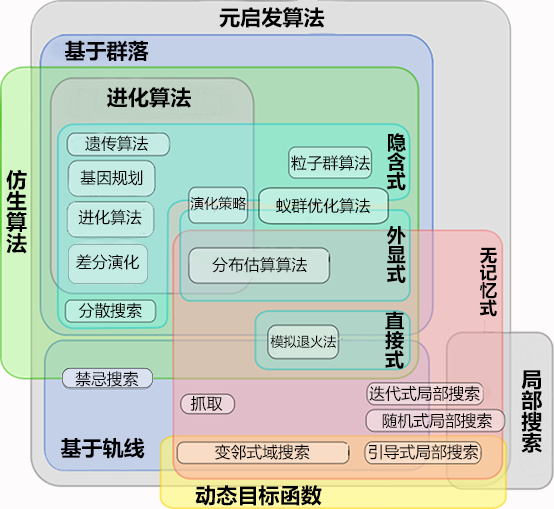


图1.1 启发式算法分类图

车辆路径问题是许多领域的热点研究问题之一，一方面是因为其拥有广泛的应用基础。比如在交通运输系统中客货运输领域。交通运输系统作为国民经济发展的支柱，其运输费用已经在国民经济支出中占有重要比重。有研究表明，在欧洲，货品的年运输周转费用大约为1680亿美元[2]。英国、法国、丹麦三国在交通运输费用上面约占国民经济支出比例为15%，9%，15%[3][4]。根据我国第三产业普查资料，我国在物流行业的成本费用之和约占国民生产总值的15%。而近几年，随着互联网的重新崛起，各个行业都面临着新的机遇与挑战。物流行业尤甚。在网络化的影响下，交通运输系统、物流配送系统，快递收发系统等行业在实时性与高效性方面开始面临更严苛挑战。物流配送路径优化，即车辆路径问题，是当今物流配送优化中关键的一环，也是电子商务活动不可缺少的内容。运输路线是否合理直接影响到配送速度、成本和效益。选取恰当的车辆路径，不仅可以加快对客户需求的响应速度，提高服务质量，还可以增强客户对物流环节的满意度，降低服务商运作成本。在国内，以阿里、京东为代表的电商企业都开始了新一轮“圈地”运动，意图建立自己的物流配送中心。以阿里为例，其最近的对海尔旗下日日顺物流进行了高达28.22亿港元的投资。而京东在物流方面的费用也逐年递增，2009年到2012年，京东物流费用率依次为4.9%、5.6%、7.2%、7.4%；2013年前三季度，该占比降为5.8%。而横向对比，苏宁的该项费用率为5.32%，亚马逊的该项费用率也为11.9%。因此，随着传统销售渠道的改变以及新的销售渠道的崛起，作为物流运作的核心，车辆路径问题（VRP）在实际应用领域的研究将变得非常重要。此外，在传统工业，比如煤炭，木材，零部件的运输方面，VRP问题的研究也有着深远的影响。

VRP问题受到越来越多的学者和研究人员的重视的另一个重要原因是其求解的计算复杂性。由于VRP是一个典型的NP难题，随着客户访问点的增加其解的规模也呈指数增长。分支定界法、动态规划等精确算法在客户访问点超过150个后会出现指数爆炸的问题。而在实际应用过程中，随着物流行业快速的发展，大规模的客户访问点的存在几乎不可避免，所以运用启发式算法是求解大规模车辆路径问题的重要途径。故本文将重点研究利用基于局部搜索的启发式算法解决带有容量限制的路径优化问题。模拟退火算法相对其他智能算法，比如遗传算法，禁忌搜索算法等在解决大规模VRP问题时具有收敛速度更快，寻优性能更好的优点，因此本文主要研究基于模拟退火算法的启发式算法及其改进算法在VRP问题中的应用。

综上所述，随着网络化的发展，计算机技术的革新以及物流对人们生活的影响不断加大，研究更加复杂更加具有实际意义的VRP问题，改进或者发现新的可以构建高质量并且有良好鲁棒性的解的算法具有重要的理论意义和实践意义。

## 1.2 论文研究的目的与意义

### 1.2.1 研究目标与内容

本文要研究是基于局部搜索的改善算法。作为元启发算法的一种，局部搜索是基于贪婪思想持续地在当前解的领域中进行搜索。在VRP问题中，局部搜索是在已有路径方案的基础上对路径在一定范围内进行变换，变换的范围即为邻域。在变化的过程中用花费更少的方案代替当前方案。直至在邻域内找到最小值。但局部搜索存在两个缺陷：一是解的质量易受到初始解和解的结构的影响，一是搜索容易陷入局部最小值而无法实现全局最优。为了克服这些缺陷，研究人员提出了许多改进算法。本次试验主要通过对模拟退火、禁忌搜索，引导式局部搜索等算法思想的掌握和理解，提出改进算法，并通过C++编程实现。

本文的研究内容主要有如下三点：

1）研究带容量约束的VRP问题研究现状；

2）设计和开发可应用于实际大规模路径优化问题的启发式算法；

3）开展计算实验，对基准数据进行测试，分析和比较设计内容2中开发算法的优劣。

论文全文共分五章，总体上的组织结构可分为三个部分：第一部分给出VRP问题的研究背景并对其研究现状进行综述；第二部分为论文的核心部分，主要研究CVRP的数学模型，三种启发式算法及改进的智能混合算法；第三部分为实验数据的分析以及讨论并对本次论文进行总结。

### 1.2.2 研究意义

首先，对车辆路径问题的研究具有重要的理论意义。VRP问题是组合优化领域中著名的NP难问题，其研究结果在物流配送系统、运输系统、快递收发系统等领域均有广泛的应用。现实生产和生活中，飞机航班安排、铁路车辆编组、邮政投递问题、码头调运、货车车队规划、水运船舶调运、燃油运输、公共汽车调度问题以及电力调度问题等都可以抽象为VRP问题来进行解决。因此，对VRP问题及其应用的深入研究，有较高的科学意义及工程应用价值。其次，对车辆路径问题的研究具有重要的现实意义。现实生活中还有很多问题可以看作车辆路径问题,其中物流配送的车辆优化调度是车辆路径问题的一个典型应用,建立合理的配送路线能有效减少商品成本,提高企业竞争力,也有利于减少物流配送引发的交通问题如：交通阻塞、交通事故、运输工具尾气污染等。

## 1.3 论文创新点

论文的创新点主要由以下几个方面：

1）通过对研究不同温度下是否有距离约束对解的质量的影响，提出了对模拟退火法的改进。根据是否有距离约束而选取不同的温度，改善解的质量的同时增强了解的稳定性。

3）使用智能的迭代方法，在两阶段法的基础上引入随机因素。第一阶段使用禁忌搜索与模拟退火法的混合算法进行diversification，第二阶段在第一阶段找到邻域最小值的基础上在该解附近进行使用下山法进行intensification。算法在两处引入随机因素，一是禁忌搜索中候选解的选取中采用grasp的策略，二是使用ClarkWright进行初始解的生成时随机产生λ的值。

2）将模拟退火法与禁忌搜索进行混合，提出智能的混合搜索算法。并使用数值进行对比，验证了算法的有效性以及改进性。

# 第2章 车辆路径问题研究综述

## 2.1 国外路径优化问题研究现状

VRP问题是组合优化领域被最广泛研究的问题之一，它在1959年最先由Dantzig and Ramser提出[5]。该问题有很多种变体，最常见的是带容量约束的VRP问题（CVRP）以及带有时间窗的VRP问题（VRPTW）。其后，人们将更多地注意力放在对VRP的复杂变体的研究上，其被称为“丰富的”VRPs(“rich” VRPs)。这些问题更接近于实际分销问题，在具体问题中可能会有多个配送中心，每辆车可能会执行多条不同的路线，车的类型也会有所不同等等情况。而近些年，由于信息和通信技术的进步使得信息可以被实时获取并处理，有关动态车辆路径问题（DVRP）的研究开始兴起。图2.1列出了与VRP问题相关的研究方向：

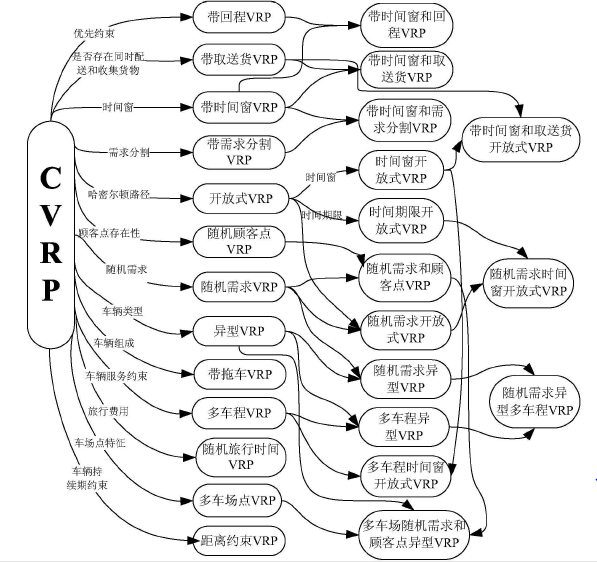


图2.1 VRP问题相关研究方向

车辆路径问题的求解方法一般可分为两类：精确算法与启发式算法。下面分别对其研究现状进行介绍：

1）精确算法：

由于最初VRP问题涉及结点较少,规模也较小,所以众多学者关注于对精确算法的研究,提出的主要精确算法有:割平面法、动态规划法、K度中心树算法、三下标车辆流、二下标车辆流、分枝定界法等。近期众多国外学者的研究成果按时间顺序可列表2.1如下。

表2.1 近期国外学者在精确算法领域的研究成果

|  |  |  |
| --- | --- | --- |
| 时间 | 作者 | 成果 |
| 2007 | Ropke,Cordeau,Vigo | 提出用于解决带有二维运输限制的CVRP问题的分支定界定价法（Branch-and-Cut-and-Price）[6] |
| 2009 | Qureshi,Taniguchi,Yamada | 提出一种新的列型分代收集精确优化算法（column generation-based algorithm）可以求解带有半软时间窗的VRP问题。[7] |
| 2010 | Gutiérrez-Jarpa, Desaulniers等人 | 提出了用于解决带有时间窗的复合车辆路径问题的算法。[8] |
| 2010 | M. Gendreau,J.-Y. Potvin | 提出一种新的列型分代收集精确优化算法（column generation-based algorithm）可以求解带有半软时间窗的VRP问题。[9] |

精确算法诞生较早,是基于运筹学的优化算法,可以求得最优解。但是该方法只适用于小规模VRP的求解,并且通常只针对特定问题才能进行设计。如果问题规模扩大,该方法的计算量将呈指数形式爆炸增长。因此当VRP规模增大时,精确算法不具有实用性。

2）启发算法：

随着VRP问题研究的深入,VRP问题涉及的因素越来越多,分类也越来越细,由此目标函数以及约束条件也越来越多。有些约束条件数学构造虽然简单,但实际求解极为困难,已被证明为NP难题。因此,近年来专家们开始关注启发式算法的应用。

传统启发式算法主要包括有:节约法、扫描法、插入法、最邻近法、两阶段法等。目前,传统启发式算法在VRP问题中已有广泛应用。传统启发式算法相比于精确算法,不再局限于小规模VRP问题的求解,应用范围更广,运算时间和复杂度也大大降低。

20世纪70年代以来,随着仿生学、遗传学和人工智能科学的发展,许多学者开始将智能算法应用到优化领域以解决相关问题。关于车辆路径问题的启发式算法的研究在90年代左右兴起，早期的算法很零散，并且主要集中在对禁忌搜索的研究上（比如1993的Taillard，1994的Gendreau，2003的Toth and Vigo[10]），不过近些年研究开始趋于理性。最好的智能启发算法是那些在搜索空间上及有深度也有广度，同时可以解决该问题的多种变体。他们既可以用于几种变换中，比如用于可调式的大规模邻域搜索（Pisinger and Ropke，2007年提出[10]），也可以用于将局部搜索和基因搜索结合的问题，比如混合遗传算法（Vidal 在2010年提出[10]）

常见的智能算法有:禁忌搜索算法(Tabu Search,TS)、遗传算法(Genetic Algorithm,GA)、模拟退火算法(Simulated AnneaHng,SA)、蚁群算法(Ant Colony Algorithm,ACA) 等。1986 年,Glover首先提出禁忌搜索算法,其思想是全局逐步寻优。而Gendreau等人最先将该算法应用到VRP的研究中,随后,Renaud、Willard、Barbarosoglu、Taillard、Duhamel、Philippe等人利用该算法求解不同的VRP问题,取得了大量研究成果。模拟退火算法最早由Kirkpatrick应用于组合优化领域。该算法是基于蒙特卡洛迭代求解策略的一种随机寻优算法,其出发点是基于物理中固体物质的退火过程与一般组合优化问题之间的相似性.[1]最近几年,Teodorovic 、Laporte通过将原子获得能量等同于分配最优结点,将原子振动等同于随机搜索线路的寻优空间，利用模拟退火算法求解VRP问题。遗传算法(Genetic Algorithm)最初是由美国Michigan大学J.Holland教授于1975年第一位提出,它是一种模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型,是通过模拟自然进化过程搜索最优解的方法。Blanton和Wainwright、Potvin、Thangiah、J.Lawrence等人都曾利用该算法解决了有时间窗口的VRP问题, Cheng和Gen利用该算法解决了模糊VRP问题。蚁群算法是由Dorigo等人首次提出,该算法是一种新型仿生类算法,其由蚂奴觅食原理的搜索机制设计而成。近期国外学者的研究成果按时间顺序可列表2.2如下。

表2.2 近期国外启发式算法研究现状

|  |  |  |
| --- | --- | --- |
| 时间 | 作者 | 成果 |
| 2001 | Berger,Barkaoui等人 | 提出了用以解决带有时间窗的路径优化问题的混合搜索算法。 |
| 2003 | Hoong，Sim等人 | 提出了路径导向的混合遗传算法用以解决带有时间窗的路径优化问题。 |
| 2005 | Montemanni，Gambardella | 使用蚁群系统解决动态车辆路径问题。[11] |
| 2005 | Y.-J.Cho，S.-D.Wang | 提出了一种阈值接受的元启发算法用以解决带有取回策略和时间窗约束的车辆路径问题。[12] |
| 2006 | Gandreau, Guertin, Seguin等人 | 提出了基于Ejection chain的局部搜索算法用以解决带有取送要求的实时DVRP问题。[13] |
| 2009 | Yuichi,Olli | 提出了一个强大的路径最小化的启发式算法用来解决带有时间窗的车辆路径问题。[14] |
| 2011 | G.-N.Abel,Bullinaria John | 提出了改进的多目标进化算法用以解决带有时间窗的车辆路径问题[15] |
| 2011 | K.-W.Peng | 提出了可调式的并行路径构建启发式算法用来解决带有时间窗的车辆路径问题[16] |
| 2011 | S.R.Balseiro,I.Loiseau and J.Ramone | 提出了蚁群算法与插入法混合的算法用以解决带有时间窗的依赖时间的车辆路径问题。[17] |

## 2.2 国内路径优化问题研究现状

近些年,随着物流行业的迅速发展,国内许多学者开始研究VRP问题。国内VRP问题受到越来越多的关注,虽然国内学者对VRP问题的研究时间较短,但是已取得很多很好的研究成果。现阶段,针对VRP问题,学者们主要通过改进各种智能算法或者将多种算法取其优点相混合的策略来实现具体的求解。研究成果按时间顺序可列表2.3如下。

表2.3 近期国内启发式算法研究现状

|  |  |  |
| --- | --- | --- |
| 时间 | 作者 | 成果 |
| 2000 | 肖鹏等人 | 构造VRP的染色体表达，采用基因换算子进行染色体重组，实现了单亲遗传算法。[18] |
| 2000 | 李军 | 设计了基于自然数编码的遗传算法，并将其用于求解非满载的VRP问题。[19] |
| 2000 | 袁健等人 | 采用神经网络求解了VRP问题[20] |
| 2000 | 祝崇俊。刘民、吴澄等人 | 通过建立三下标流模型，同时结合2-opt算法解决了较大规模的VRP问题。[21] |
| 2001 | 刘浩等人 | 应用模拟退火算法求解了两车型随机需求的VRP问题。[22] |
| 2002 | 张丽萍等人 | 通过引入新颖交叉算子，构造了一种改进遗传算法，该算法可以有效求得VRP的优化解。[23] |
| 2002 | 郎茂祥 | 利用遗传算法求解了一般VRP问题。而后其又改进了传统的遗传算法以解决对VRP问题搜索效率低等缺陷。[24] |
| 2004 | 崔雪丽等人 | 基于人工蚂蚁系统，结合2-opt交换策略局部搜索机制，给出了可快速求解VRP问题的蚁群算法。[25] |
| 2004 | 刘云忠等人 | 应用蚁群算法求解了VRP问题。[26] |
| 2005 | 张建勇等人 | 运用混合遗传算法结合模糊可能性理论讨论了需求不确定的VRP问题。[27] |
| 2009 | 崔雪丽等人 | 设计了混合改进型蚂蚁算法来求解VRP问题。[28] |
| 2011 | 于斌等人 | 设计了一个蚁群最优模型，用以解决带有时间窗的路径优化问题。[29] |

近期在该领域的成就主要有以下几个方面：1）对取送问题进行了优化，在该问题中，货物从供应商处被运输至分销中心，在交接设施处进行运输整合。其用美国的一个大型零售商的数据进行了测试。2）为各种路径问题引入了模型和有效不等式。该应用主要起于外勤工作和家庭护理交付3）概率型的带有时间窗的TSP问题，其中顾客的出现为概率事件。作者提出了一个资源模型以及一个可变的邻域递减搜索4）引入一个军用航空器计划问题，其中一对飞机要攻击许多地面目标。问题可被归纳为带有同步和优先约束的VRP问题。它可由线性规划和整数规划的混合来表示。在实际中可用Cplex求解。[10]

随着人们对 VRP问题研究的深入以及对 VRP问题解的质量要求的提高，人们开始研究如何在算法中加进人的主观判断以提高解的质量，比如如何在行驶过程中判断到仓库补货的时机，这归结为补货策略问题；另外，人们也开始研究如何结合顾客库存的情况来制定运输策略的问题，这归结为库存运输问题；诸如此类的研究是当前研究的重点。

# 第3章 算法及数据结构

## 3.1 现代启发式算法

根据Stutzle的定义，现代启发式算法是指一种典型的高级策略，该策略引导基本的问题使用特定的启发式算法来增强其性能。大多数现代启发式算法的搜索过程是一种概率决策过程，但是这种搜索与纯粹的随机搜索的最大的不同在于它不是盲目地随机搜索，而是以一种智能的形式进行随机搜索。[30]

有上述定义可以看出，现代启发式算法是一种利用一组不同的启发式算法对搜索空间进行探索的高级策略。它的本质思想就是根据不同的问题实现多样化搜索（diversification）和集中搜索（intensification）之间的动态平衡。多样化搜索是指在整个搜索空间进行大范围的搜索，而集中搜索则是指充分利用多样化搜索是中搜集到的信息对特定的区域进行深度挖掘。这种策略的优势在于，既可以快速的探索包含高质量的解的区域又不必浪费太多资源在探索过的区域或者不太可能包含高质量解的区域进行探索。下面综述一些具有代表性的启发式算法。包括模拟退火法、禁忌搜索、引导式邻域搜索。其中的模拟退火法为本实验的基础，并在此基础上运用禁忌搜索与模拟退火法的结合对问题进行求解。

### 3.1.1模拟退火法

模拟退火是一种贪心算法，但是它的搜索过程引入了随机因素。以一定概率接受劣解是其最重要的特性。模拟退火算法与爬山法类似，但是它没有选择最佳的移动，而是选择随机的移动。如果该移动使情况得到改善，那么接受该移动；如果并没有改善，算法以某个概率接受该移动，概率的计算方式为：exp(-Δt′/T)。模拟退火算法要求在算法进行的过程中，逐步降低其跳出局部最优的概率，使其越来越趋于稳定，最终达到全局的最优解。图3.1给出了模拟退火法算法框架。

模拟退火法的主要过程为：在由ClarkWright初始化的解的基础上对每一个点在其邻域内依次进行三种进行邻域搜索，每进行一次搜索则首先判断当前解是否满足约束，若满足约束则再判断当前解的节约值是否小于0，若小于0则将该次移动标记为可行，若大于0则以一定概率决定是否将该移动标记为可行。概率公式为为：

exp(-Δt′/T) > random[0,1） (3.1.1)

其中Δt′表示节约值，T表示当前温度。从该公式可以看出，当T越高或者当前解与最优解的差距越小则接受劣解的概率越大。图3.2中的参数和函数需要根据问题具体设置。

· 函数**CenerateInitialSolution**产生初始解，本文通过ClarkWright节约法进行初始解的构造

* 函数**InitializeAnnealingParameters**初始化退火过程中的参数值。其中包括（1）初始温度；（2）算法在每一个温度迭代的次数；（3）算法外部循环的终止条件
* 函数**UpdateTemp**更新温度值，本文使用特定退火率对温度进行更新。
* 函数**GenerateNeighbor**从当前解s的邻域产生新解
* 函数**AcceptSolution** 根据式（3.1.1）对解进行接受。

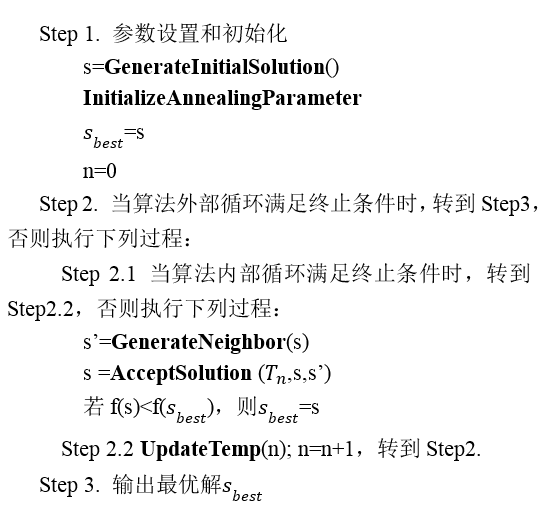


图3.1 模拟退火法算法框架

### 3.5.2 禁忌搜索

禁忌搜索是一种带有记忆功能的启发式算法，它使用一个禁忌列表来记录最近实施的搜索过程，通过将局部最优解的某些特征列为禁忌从而在接下来的若干步迭代中禁止触碰带有这些特征的解。该长度成为禁忌长度。这样就避免搜索多次回到同一解上，以便提高算法跳出最优解的能力。但当搜索到优于当前最优解的解时则破除禁忌，从而提高搜索效率。在选择禁忌对象时，禁忌搜索会形成一个候选解集，从该解集中选择节省路径最多而没有被禁忌的对象。图3.2给出了简单的禁忌搜索的算法框架：

本实验采用的禁忌搜索为，由ClarkWright 产生初始解S，在初始解的基础上进行邻域搜索，生成当前解的所有邻域解{Si}，然后根据节约值排序选出最优的五个候选解{Si’}。对候选解进行判断：1）是否被禁忌2）是否满足蔑视准则。没有被禁则接受该解，并将该解的禁忌元素添加入禁忌列表中。若被禁则检查是否满足蔑视准则。若满足蔑视准则，则将该解作为当前最优解并破除禁忌，重新将该解的禁忌元素添加入禁忌列表。若不满足则不接受该解。

禁忌对象：

算法中禁忌对象指的是禁忌表中被禁忌的那些变化的元素。本实验将每次移动影响到的路径作为禁忌对象。

禁忌表的设计：

本实验的禁忌表中的元素为被禁忌路径r对应的hash值。每次检查当前路径是否为禁忌时，只需检查该路径的hash值是否已存在与禁忌列表中。Hash值由一个hash函数产生。产生步骤如下：

1. 检查路径的是否为有向的（即路径中第一个客户点的索引值小于最后一个的，否则对路径进行颠倒）
2. 将路径中相邻的两点的索引值相加产生一个新的数组。比如：路径r：0-2-4-5-6-0。经变换后为{6,9,11}
3. 选择三个随机数与这三个数相关联：{Y[6],Y[9],Y[1]}
4. 做XOR和（Y[6]Y[9]Y[1]）
5. 取该值的对数后得到hash值。

禁忌表长的设计：

禁忌表长是Ts中的一个关键因素。一般根据实际问题的不同而不同。表长过短会导致搜索过程中出现循环，过长会导致太多的禁忌，降低搜索的作用，本文后面的实例取禁忌长度为3。

候选解：

候选解为每个点邻域搜索后得到的5组最优移动。

设置参数，产生初始解

开始

是否满足终止条件

输出最优解

生成当前解的邻域解，选出候选解

是否满足蔑视准则

将满足蔑视准则的解作为当前解，用它包含的禁忌元素替换最早进入禁忌表的对象，更新最优解

将非禁忌的最优候选解作为当前解，用该解包含的的禁忌元素替换最早进入禁忌表的元素

Yes

No

Yes

No

图3.2 禁忌搜索算法框架

### 3.5.3 引导式邻域搜索

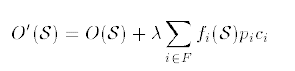
引导式局部搜索是基于惩罚的通用启发式算法。该算法的工作机理是，基于搜索获得的经验为目标函数增加一个惩罚系数。大致来说，如果搜索游离到距离之前已被访问的局部最小值太近的话，它将会被惩罚。

引导式局部搜索通过惩罚特殊解的特征从而移出局部最小值，该特征被认为不应该出现在近似最优解中。该算法定义了一个被修正的目标函数，该函数带有一系列与解的特征有关的惩罚项的参数。然后我们会调用传统的局部搜索改善参数目标函数。

引导式局部搜索需要如下部分：

* 一组特征F。对于特征i∈F，令该特征的指标函数为fi。如果特征i在解S中，则令fi(S)=1，否则令fi(S)=0。
* 一个成本矢量为c，ci表示特征i的成本。
* 一个惩罚系数λ。本实验中λ取值为0.2。

引导式局部搜索通过一个惩罚矢量p追踪被执行的惩罚。Pi是特征i目前为止被惩罚的次数（整数）。假设O(S)是问题的原始目标函数，引导式局部搜索定义如下参数目标函数：



在本实验中，被惩罚的特征元素为边，构建一个惩罚矩阵p[i][j],为每条边的惩罚项。每确定进行一次移动，便将该次移动所涉及的边的惩罚项加1。在进行节约值的比较时，使用带有惩罚项的节约函数：Saving()。即每条边的距离值变为：d[i][j]+0.2\*d[i][j]\*p[i][j]。

## 3.2 数据结构

### 3.2.1 数据类型

**VRP**

该类为该程序的主类。

该类中包含程序调用内部数据的函数以及进行算法的主要函数。

包含描述一个VRP问题的各个数据：已知最优解，客户点数，每个点的坐标，每个点之间的距离矩阵，各个点的需求，容量限制以及路长限制；包含进行算法时需要的数据：最优解和当前解的缓存，邻域大小以及邻域矩阵，模拟退火法中的温度及退火率；其他类：VRPMove，VRPNode，VRPRoute，VRPTabulist等。

**VRPMove**

该类用于记录每次移动后的信息。

包含一个用于比较两次移动的函数，该函数用于实现BestAccept策略。

包含的数据有：被影响的路径总数，路径编号，路径路长，路径需求，路径客户数，节约值，总路径个数，移动后路径长度，移动类型，受影响的参数的个数等。

**VRPNode**

该类用于记录每个点的信息

包含点的坐标值，点的编号，点的需求量，点的邻域。

**VRPRoute**

该类用于记录路径的信息

包含一个哈希函数，用于生成一条路径的哈希值，该哈希值被用于禁忌搜索。

包含一条路的起点、终点、长度、容量、客户数。路径中客户点的排列顺序，该路径的哈希值。

**VRPTabulist**

该类用于禁忌搜索算法。

包含禁忌列表类以及候选解类。

包含更新禁忌列表的函数，清空禁忌列表的函数，更新候选解以及清空候选解的函数。

包含哈希值，VRPMove类等。

### 3.2.2 解的存储方式

当前解被储存在一个double linked list中，其由pred\_array[ i ]和next\_array[ i ] 构成。这一对数组中的元素为指针，通过将pred\_array[ i ]和next\_array[ i ]的指针指向特定的node来储存一个解（即路径中节点的连接顺序），其中negative indices 表示一个新的路径的开始。可由图3.3和3.4说明：

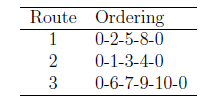


图3.3

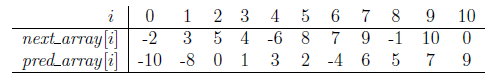


图3.4

### 3.3.3 哈希表示

本程序中的禁忌搜索使用一对哈希值表示一个禁忌元素，这样既节省了存储空间同时可以快速查看某元素是否被禁忌，加快了搜索效率。该方法的主要思想是，利用一个“hash\_function”将一个解转换成一个“标准形式”再将其存入一个缓存中，最后将该缓存与一个整数关联。已知我们有一个由n个点的VRP问题产生的具有R条路径的解。对于1至R的每条路径j，我们用一个有序列代表第j条路。此外，我们储存了一列n个随机32位整数，,,…,。然后通过执行“hash\_function”我们将每一个解同一个正整数关联。图3.5表示如何利用一个“hash\_function”将解转换成一个标准形式。

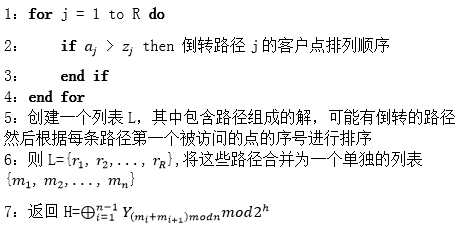


图3.5 将解转换成一个标准形式

在步骤2中，我们保证每条路都是有向的以便每条路的第一个点的序号值小于该条路中最有一个点的序号值。然后我们在步骤5，6将这些路径合并成一个单独的列。在步骤7中，我们将序列表中的每对点与一个32位的整数相关联并且将他们加和。（我们没有使用正常的加和操作而是按位加和XOR）

### 3.2.4 各模块之间的关系

将最优解导入VRP中

检查禁忌状态

**VRPTabulist**

禁忌列表，禁忌元素的哈希值，候选解，以及相关函数

禁忌列表更新至VRP

**LocalSearch**

OnePointMove

TwoPointMove

Two-Opt

**VRPIO**

读取TSPLIB文件函数

写出解文件函数

导入到解缓存函数

从解缓存导出函数

节点数，节点间距离，约束，各点需求等。

受影响的路径，移动类型，移动 节约值等

最优解和当前解缓存，搜索空间矩阵，搜索规模，禁忌列表等。

**VRP**

读取函数

**VRPMove**

搜索（Search）一次移动（Move）

评价（evaluate）一次移动

进行移动（Move）

**Solver**

根据不同的算法策略，调用局部搜索，采用一定的迭代策略对算法进行实现。包括模拟退火法和智能混合算法。

调用局部搜索

Update\_Move函数将VRPmove中的信息更新至VRP

生成禁忌列表

**VRPRoute**

路径起点，终点，路长，路径运输量，路径编号，路径中包含的客户数

图3.6 各模块间关系

## 3.3 本章小结

本章主要分为两部分，分别对现代式启发算法和本实验所使用的程序的数据结构进行了说明。

在第一部分中，详细介绍了本文会用到的算法：模拟退火法及禁忌搜索。并对引导式邻域搜索进行了简单的介绍。在第二部分中，说明了本实验使用的程序的数据类型，解的存储方式，禁忌搜索使用到的哈希表示。并通过图形的方式说明了程序各个模块之间的关系。

# 第4章 带容量约束的车辆路径问题及算法

## 4.1 问题提出

带容量限制的车辆路径问题，即CVRP。在CVRP问题中，与每位顾客对应的运输量以及每位顾客的需求都是确定的，或者说已知的，而且不能被切分。每辆车都是相同的并且从同一个配送中心出发，而且每辆车只有容量限制。目标是满足所有顾客的需求并且使得总花费最小。

CVRP问题可以被描述为一个图论问题。设G=(V, A)是一个完全图，V={0,…,n}为点集，A为弧集。点集的点对应顾客，0点为配送点。定义一个非负花费，其为在其对应弧集A中的一条弧(i, j)中，从点i至j的花费，一般为从i至j的距离。花费矩阵必须满足三角不等式：

+ ≥ 对于所有的i,j,k ∈V （4.1.1）

车辆具有的最大装载能力为C，有N位顾客，每位顾客有其需求量D。车辆从配送中心出发对客户进行配送服务最后返回场站，要求所有顾客都被配送，每位顾客一次配送完成，且不能违反车辆容量的限制，目的是使所有车辆路线的总距离最小。

本章使用模拟退火法对CVRP进行求解，并提出了一个集成禁忌搜索及变温模拟退火的智能混合算法。本章的结构如下：4.2 建立问题的数学模型并定义模型中涉及的参数。4.3 详细给出了本章所提出了智能混合搜索算法。4.4 给出算法的计算结果并与模拟退火法进行比较。4.5 对本章进行了总结并探讨了未来的研究方向。

## 4.2 数学模型构建

### 4.2.1 问题假设

1）每个需求点的需求已知

2）每个需求点的位置已知

3）需求点之间的距离按直线最短距离计算

4）每个需求点只由一辆车服务一次

5）每辆车由配送中心出发再回到配送中心，且每辆车的最大容量相同且已知。

### 4.2.2 参数定义

该问题一半由图论定义。G=(V,A)为一个完全图，V是点集，代表顾客和总站，A是弧集代表路径。表示从顶点i到顶点j的花费（该花费一般代表两点之间的距离）。表示每个顾客的需求。对于一个包含于V的点集S，令d(S)表示S中所有顾客的需求。K表示车辆数，C表示车的容量。表示需要服务所有顾客的车辆数。r(S)为需要服务S中所有顾客的最小车辆数。这样CVRP即为寻找最小花费的前提下满足所有顾客的K个简单圈。并有如下约束：

·每个圈必过总站点

·每个顾客访问点只被一个圈访问

·一个圈的顾客总需求不能超过车辆容量。

### 4.2.3 模型建立

基于上面的设置，可以得出CVRP的基本模型，为0-1整数规划。定义变量：

其数学模型为：

 (3.1)

Subject to

  (3.2)

  (3.3)

 (3.4)

 (3.5)

  (3.6)

  (3.7)

(3.1)目标函数表示令每条路的花费的总和最小。

(3.2)(3.3)表示从每个点进入与出去的选择只有一个。

(3.4)(3.5)表示从总站进入的与从总站出去的必须等于总车辆数。

(3.6)表示对于某点集S，穿过该集的弧的个数一定要不小于服务该集的最小车辆数。（容量切割限制）

## 4.3 CVRP的智能混合算法

由于单纯使用模拟退火所花费的求解时间过长，而单纯使用禁忌搜索容易陷入局部最优解，故本算法将两种算法相结合，在扩大邻域搜索范围的同时避免进行迂回搜索从而实现既提高解的质量又提高求解速度的目的。同时为了增强解的质量，该算法同时使用了两阶段法。即用来diversification的uphill阶段以及用来intensification的downhill。其核心思想是，在爬山阶段使用混合算法接受一些劣解并以最大效率搜索解空间。搜索完毕找到局部最优解，在该解附近进行下山法的搜索。在多次尝试后无法跳出特定的局部最小值后算法停止。算法需要设置如下参数：

* D：控制在diversification阶段的主循环规模。经参数实验发现该值在25至50得到的结果较好，本实验选在30。
* K：尝试突破局部最小值的次数。
* N：在执行局部搜索时邻域的规模，该值过大会导致算法求解速度变慢

算法框架如图4.1。

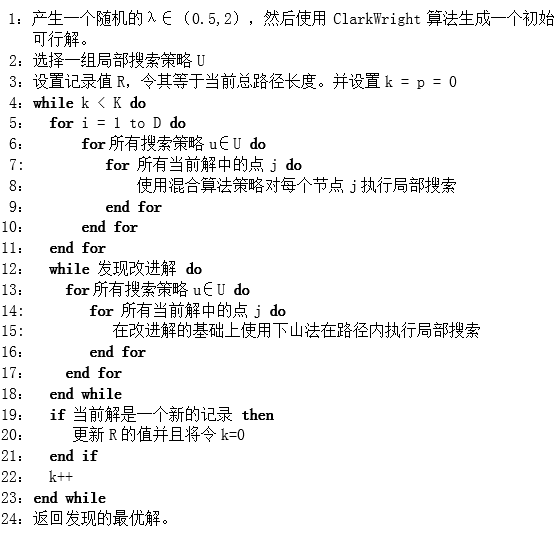


图4.1 智能混合算法伪码

### 4.3.1 初始解的构建

该方法用于建立初始解。该算法的核心思想是节约算法，即将两条路径合并为一条从而“节约”路径。如图4.2.

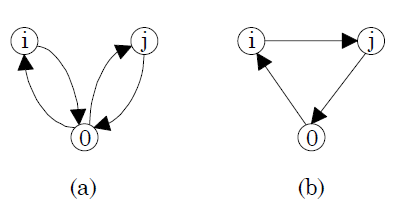


图4.2 节约法

(a)方案的花费为：C0i+C0j+Ci0+Cj0

(b)方案的花费为：C0i+Cj0-Cij

节约值Sij=Ci0+C0j-Cij

其步骤大致如下：

首先，计算每个客户点之间的节约值savings，然后按照降序排列。接着从序列顶部抽出一对点进行检查，要保证如下两点：

1. 将i-j相连后不会删除原先已经建立的客户点之间的联系，即i，j点必须在路径的端点。
2. 不得超出容量限制和距离限制。

上述步骤完成后，重新进行排列并检查。算法框架如图4.3所示：

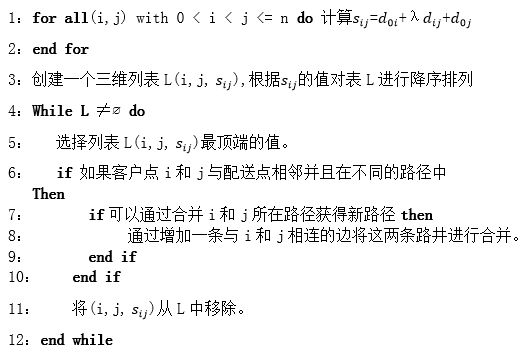


图4.3 ClarkWright算法框架

图4.4、4.5、4.6展示了该算法不同阶段的解

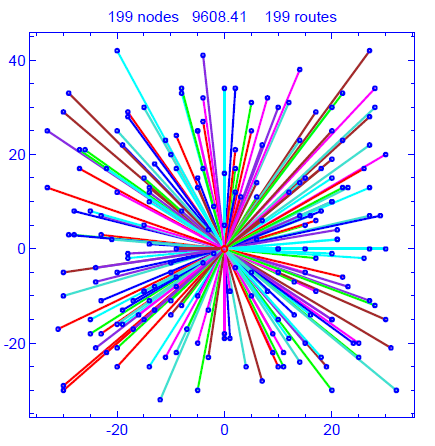


图4.4 初始解

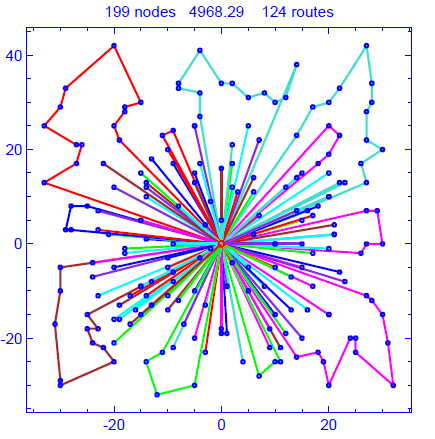


图4.5 合并100次之后

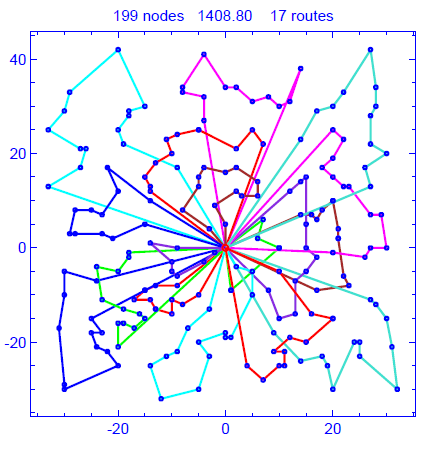


图4.6 最终解

### 4.3.2 局部搜索

局部搜索时组合优化问题中最常用的技巧之一。所谓局部搜索，其实质是变换。是指使用特定的操作方式对一定区域内的元素进行变换。对于VRP问题来说，所谓的邻域实际上是跟据局部搜索策略决定的，即解中的一个点被移动至一个新的位置或者几条边被移除或者被新边取代。换句话说，对于特定的解和局部搜索策略，邻域即对当前解使用局部搜索策略所能得到全部解。局部搜索已被证明对处理VRP问题十分有效，并且这项技术已被用于许多VRP元启发算法中。

比较常用的局部搜索策略有有2最优，交换，单点移动，两点移动等等。通过比较搜索前后的解决定是否接受该解，若接受则再在该解的基础上进行下一步搜索。但是这种方法最大的缺陷就是容易陷入局部最优解，即搜索过程陷入某种循环。为此，需要使用其他算法对其进行改进。本研究使用的局部搜索有单点移动、两点移动、2最优移动。

对于VRP问题，大量文献证明：在局部搜索中加入禁忌搜索策略可大幅提高搜索效率。[ ]因此在图4.1中混合算法局部搜索阶段，考虑使用禁忌搜索算法。禁忌搜索是由Glover提出的一种带有记忆的局部搜索策略[]，它利用这种记忆式算法跳出局部最优解。本算法使用具有短期记忆的禁忌搜索。本算法的禁忌搜索阶段包含以下几个重要组成：

（1）初始解 为了开始进行搜索，禁忌搜索需要一个初始解，本实验使用由ClarkWright产生的解作为初始解。

（2）解的评价 本实验以模拟退火机制对解进行接受。

（3）邻域结构 本实验在执行禁忌搜索算法时，对当前解依次使用三种算子来产生邻居解。即2-opt，单点移动，两点移动。具体操作如图4.7所示：

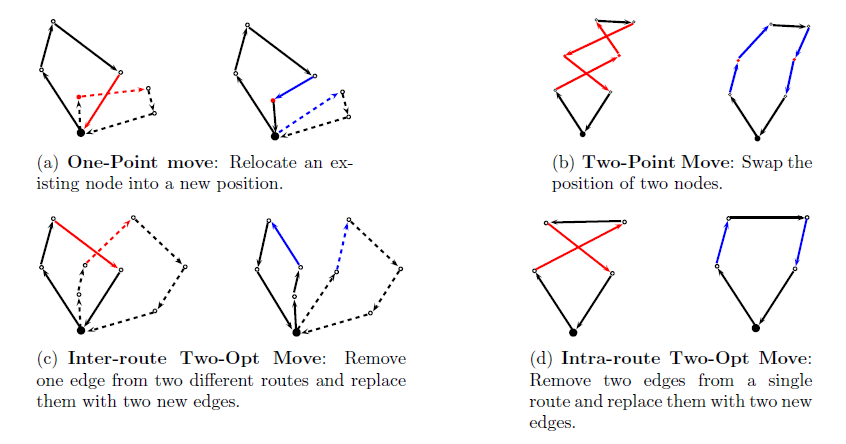


图4.7 局部搜索算子

（4）禁忌对象和禁忌表 本实验将路径作为禁忌元素，即将每一个被接受的解中受影响的路径作为禁忌元素加以禁忌并记录在禁忌表中。根据文献[ ]所述，这里采用随机禁忌表长度。禁忌表的长度在[5,10]之间随机选取。

（5）特设准则 在禁忌搜索过程中，可能出现全部邻居解被禁或者目前为止的最优解被禁的情况，因此本算法设置以目标函数值为判断标准的特设准则。

### 4.3.3 退火机制

本算法的退火机制以传统模拟退火法为基础，根据温度进行的实验得出的结论将其改进为变温模拟退火法。即当有距离约束时将温度设置为6，没有距离约束时温度设置为2。每进行一次完整迭代对温度进行降温，降温速率为0.99。

### 4.4.4 集中搜索过程

本算法在主循环结束后，会执行一个集中搜索过程用于对解进行后期优化。该过程采用下山策略，即只接受优于前一个解的解。在此过程中使用的迭代策略较为灵活，并不是固定的迭代数，而是只要每次搜索发现新的最优解便进行一次迭代直至不会发现新的最优解。

## 4.4 实验和结果

### 4.1.1 测试问题

本文采用取自TSPLIB的三类算例进行试验，分别为Christofides、Li、Goledn。TSPLIB是一个实例库，它是世界上被使用最广泛的实例库之一，由G.Reinelt在1991年提出。[]，其中有来自不同资源的不同类型的VRP实例。本文的实例取自网站TSPLIB，其中包含适用于各种TSP相关问题的实例。与程序适配的所有问题均可从网站https://sites.google.com/site/vrphlibrary/benchmark-por下载。问题的点数分布从50至1200，所有问题使用的距离均为欧几里得距离。表A1给出了测试问题的特征信息。

本文根据客户点数对算例进行了分类。共分为三类：C1为客户点数50-200，包含14个算例，其中六个有距离约束。C2为客户点数241-484，包含20个算例，其中八个有距离约束。C3为客户点数560-1200，包含十二个算例，均有距离约束。实验的所有程序均有C++程序编写。实验运行环境为Intel®Core™i5-2410MCPU@2.30GHz/4000M RAM PC。

下面分别对智能混合算法和模拟退火法进行测试。不同算法的迭代方式不同。初始解由ClarkWright 从三个[0.5,2]内随机选择的λ值生成。目标函数值为所有车辆行驶的总距离。各算法的其他参数在求解的各自部分说明。

### 4.1.2 参数设置

本文所使用的智能混合搜索算法以模拟退火法为基础，将禁忌搜索嵌入局部搜索。故需要进行一些基本的参数设置。本节通过一些基本实验实现最优参数组合。智能混合算法需要设置的参数有：

T – 模拟退火法初始温度

r – 退火率

N – 局部搜索时每个点邻域的规模

D – 控制在diversification阶段的主循环规模。

K – 尝试突破局部最小值的次数。

C – 禁忌搜索中候选解的个数

L – 禁忌长度

本算法根据来自文献[11]的经验性结论，当退火率在0.8-0.99时产生的结果较好，且该值越接近1越好。故本实验将退火率设置为0.99。经过数次参数调整实验，发现当D=30，K=5，N = 10时可以在合理地时间内得到优质的解。根据文献[12]，禁忌搜索候选解个数为5时产生的解较理想。禁忌表的长度在[5,10]中随机选择。由于初始温度不易确定，所以进行一些基本实验确定温度的最优值。

本实验分别对C1组，C2组和C3组在不同温度下进行测试。得到的结果如表A2、A3、A4所示（见附录），我们将图A1、A2、A3中有距离约束的实例标记为红色，没有距离约束的实例标记为蓝色。这样，我们从表中数据和对图的观察可以得出以下结论：

1. 从表A2及图A1可以看出。当客户点数较小时，温度的变化对解的质量基本没有影响。且没有距离约束的情况下算例的解的质量要优于距离约束存在时的算例。
2. 根据表A3，从同一温度下不同算例的均值来看，当温度在2左右时解的质量最好，而当温度不断升高后，解的质量也随之下降。而从图A2上可以看出，有距离约束的算例1,8,4,3,2,6,7,5随着温度的升高解的质量逐渐变好并趋于稳定，而其他没有距离约束的算例，随着温度的升高解的质量逐渐变坏。而温度2,3便是有距离约束和无距离约束的两类算例解的质量上升和下降的交点，所以平均来看在此处解的质量最好。
3. 由图A3可以看出，随着温度的增加，解的质量在改善并趋于稳定。并且，当温度大于5后解的均值趋于稳定。

因此我们根据上述观察对模拟退火机制进行了改进，引入了变温机制使得算法对距离约束有更好的适应性。具体来说，当没有距离约束时，温度为2，当有距离约束时，温度为6。数据如表4.5。

表4.5 引入变温机制

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Best | Name | Best | Name | Best |
| Christofied\_01 | 1 | Golden\_01 | 1.03 | Li\_21 | 1.026 |
| Christofied\_02 | 1.005 | Golden\_02 | 1.01 | Li\_22 | 1.025 |
| Christofied\_03 | 1.002 | Golden\_03 | 1.015 | Li\_23 | 1.027 |
| Christofied\_04 | 1.007 | Golden\_04 | 1.022 | Li\_24 | 1.01 |
| Christofied\_05 | 1.016 | Golden\_05 | 1.03 | Li\_25 | 1.027 |
| Christofied\_06 | 0.945 | Golden\_06 | 1.021 | Li\_26 | 1.011 |
| Christofied\_07 | 0.919 | Golden\_07 | 1.013 | Li\_27 | 1.03 |
| Christofied\_08 | 0.955 | Golden\_08 | 1.022 | Li\_28 | 1.042 |
| Christofied\_09 | 0.892 | Golden\_09 | 1.029 | Li\_29 | 1.031 |
| Christofied\_10 | 0.95 | Golden\_10 | 1.031 | Li\_30 | 1.052 |
| Christofied\_11 | 1 | Golden\_11 | 1.034 | Li\_31 | 1.04 |
| Christofied\_12 | 1 | Golden\_12 | 1.037 | Li\_32 | 1.021 |
| Christofied\_13 | 0.677 | Golden\_13 | 1.021 | AVG | 1.0285 |
| Christofied\_14 | 0.946 | Golden\_14 | 1.024 |  |  |
| Avg | 0.951 | Golden\_15 | 1.021 |  |  |
|  |  | Golden\_16 | 1.025 |  |  |
|  |  | Golden\_17 | 1.013 |  |  |
|  |  | Golden\_18 | 1.026 |  |  |
|  |  | Golden\_19 | 1.019 |  |  |
|  |  | Golden\_20 | 1.019 |  |  |
|  |  | Avg | 1.0231 |  |  |

将上述数据与恒温下的最好数据进行对比，可以发现对于C1组的算例，恒温组的平均最优解出现在温度为2时，平均最优解为0.951571。而此时变温组的最优解为0.951。对于C2组的算例，恒温组的平均最优解出现在温度为2时，为1.031。而变温组的平均最优解为1.0231。对于C3组的算例，由于该组算例均有距离约束所以平均最优解相同。

上述分析说明变温机制的引入可以改进解的质量。至此参数设置完毕。

### 4.1.3 实验结果

本算法包含三个随机因素：1）使用ClarkWright法创建初始解时λ的选择。2）产生候选解后在候选解中随机进行选择，并在此解的基础上进行迭代。3）禁忌表长的选择。由于包含随机因素，故本次实验对每组数据进行十次运算后取平均值。实验数据见表A6、A7、A8（见附录）。此外，本文在附录中列出分别列出客户点数为50、100、200、400、800、1000时，由混合算法产生的解的图像。

本文智能混合算法与单纯的模拟退火法在时间与解的质量的比较如图4.8-4.13及表4.9、4.10、4.11。

图4.8 C1组混合算法与模拟退火法在时间上的比较

图4.9 C1组混合算法与模拟退火法解的质量上的比较

图4.10 C2组混合算法与模拟退火法在时间上的比较

图4.11 C2组混合算法与模拟退火法在解的质量上的比较

图4.12 C3组混合算法与模拟退火法在时间上的比较

图4.13 C3组混合算法与模拟退火法在解的质量上的比较

表 4.9 C1组混合算法与模拟退火法的比较

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | S\_Time | ST\_Time | S\_Solut | ST\_Solut |
| Christofied\_01 | 1.29 | 0.572 | 1 | 1 |
| Christofied\_02 | 2.84 | 1.219 | 1.005 | 1.0102 |
| Christofied\_03 | 5.9 | 2.07 | 1.002 | 1.0035 |
| Christofied\_04 | 11.87 | 3.741 | 1.007 | 1.0123 |
| Christofied\_05 | 20.14 | 5.672 | 1.016 | 1.0211 |
| Christofied\_06 | 1.33 | 0.806 | 0.945 | 0.945 |
| Christofied\_07 | 3.01 | 1.732 | 0.919 | 0.9259 |
| Christofied\_08 | 5.82 | 3.534 | 0.955 | 0.9565 |
| Christofied\_09 | 12.74 | 5.352 | 0.892 | 0.8941 |
| Christofied\_10 | 21.4 | 7.249 | 0.95 | 0.9435 |
| Christofied\_11 | 7.38 | 2.897 | 1 | 1.0012 |
| Christofied\_12 | 6.08 | 2.131 | 1 | 1 |
| Christofied\_13 | 7.53 | 3.755 | 0.677 | 0.6772 |
| Christofied\_14 | 6.21 | 2.958 | 0.946 | 0.946 |
| AVG | 8.11 | 3.120571 | 0.951 | 0.952607 |
| DEV |  |  | 7.67E-03 | 7.78E-03 |

表4.10 C2组混合算法与模拟退火法的比较

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | S\_Time | ST\_Time | S\_Solut | ST\_Solut |
| Golden\_01 | 17.85 | 6.9 | 1.03 | 1.0253 |
| Golden\_02 | 27.1 | 10.858 | 1.01 | 1.0192 |
| Golden\_03 | 56.19 | 15.861 | 1.015 | 1.0238 |
| Golden\_04 | 83.86 | 20.032 | 1.022 | 1.0122 |
| Golden\_05 | 21.03 | 3.719 | 1.03 | 1.0279 |
| Golden\_06 | 34.69 | 7.411 | 1.021 | 1.0183 |
| Golden\_07 | 42.07 | 11.949 | 1.013 | 1.0193 |
| Golden\_08 | 60.33 | 16.99 | 1.022 | 1.0319 |
| Golden\_09 | 34.82 | 6.151 | 1.029 | 1.0238 |
| Golden\_10 | 56.19 | 11.314 | 1.031 | 1.023 |
| Golden\_11 | 85.75 | 13.881 | 1.034 | 1.0223 |
| Golden\_12 | 125.74 | 18.725 | 1.037 | 1.0252 |
| Golden\_13 | 33.32 | 6.469 | 1.021 | 1.0267 |
| Golden\_14 | 56.27 | 10.242 | 1.024 | 1.0258 |
| Golden\_15 | 82.85 | 14.22 | 1.021 | 1.0254 |
| Golden\_16 | 123.96 | 18.847 | 1.025 | 1.0284 |
| Golden\_17 | 31.51 | 9.106 | 1.013 | 1.0077 |
| Golden\_18 | 52.77 | 12.01 | 1.026 | 1.0207 |
| Golden\_19 | 75.44 | 15.661 | 1.019 | 1.0192 |
| Golden\_20 | 93.04 | 19.765 | 1.019 | 1.0189 |
| AVG | 59.739 | 12.50555 | 1.0231 | 1.02225 |
| DEV |  |  | 5.30421E-05 | 3.15E-05 |

表 4.11 C3组混合算法与模拟退火法的比较

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | S\_Time | ST\_Time | S\_Solut | ST\_Solut |
| Li\_21 | 201.66 | 23.565 | 1.026 | 1.0288 |
| Li\_22 | 123.16 | 28.944 | 1.025 | 1.027 |
| Li\_23 | 226.25 | 28.381 | 1.027 | 1.0207 |
| Li\_24 | 264.61 | 33.961 | 1.01 | 1.0186 |
| Li\_25 | 210.43 | 43.638 | 1.027 | 1.0267 |
| Li\_26 | 304.37 | 36.879 | 1.011 | 1.0125 |
| Li\_27 | 278.66 | 55.156 | 1.03 | 1.0341 |
| Li\_28 | 389.81 | 43.462 | 1.042 | 1.0323 |
| Li\_29 | 431.7 | 52.484 | 1.031 | 1.0095 |
| Li\_30 | 515.01 | 58.188 | 1.052 | 1.0117 |
| Li\_31 | 790.74 | 69.509 | 1.04 | 1.0352 |
| Li\_32 | 834.42 | 71.775 | 1.021 | 1.0409 |
| AVG | 380.90167 | 45.49517 | 1.0285 | 1.024833 |
| DEV |  |  | 1.46E-04 | 1.04E-04 |

从上面的图像与数据中我们可以得出以下几个结论：

1. 从图4.4-4.9以及表4.8我们可以看出，在解的质量方面，智能混合算法与模拟退火法的差距虽然很小，但是在求解时间方面，智能混合算法要远远优于模拟退火法算法。尤其当客户规模较大时，改进算法在求解时间方面的优势十分明显。
2. 通过对表4.9-4.11中方差的比较可以发现智能混合算法产生的解的稳定性更好。在C1组中，模拟退火算法的方差为7.67E-03，混合算法为7.78E-03，基本一致。在C2组中，模拟退火算法为5.30421E-05，混合算法为3.15E-05，大大优于模拟退火法。在C3组中，模拟退火法为1.46E-04，混合算法为1.04E-04。稳定性提高的原因可能与采用变温策略有关。
3. 从表4.9-4.11的结果可以看出，随着客户规模的增大，两种算法的求解时间均有所上升，解的质量有所下降。但是，在时间方面，混合算法求解时间的增加要远远小于模拟退火法；在解的质量方面，混合算法的解质量的下降程度要小于模拟退火法。

## 4.5 本章小结

在物流配送日益重要的今天，作为该问题最基本的模型，CVRP的研究是解决这类问题的基础。本章首先对CVRP的提出和数学模型进行了说明，接着提出了一种智能混合算法，并在实验的基础上对其中的某些参数进行了调整与设置。之后通过与相同条件下由模拟退火法产生的结果的比较，验证了算法的有效性与稳定性。此外，在参数设置阶段发现了距离约束与模拟退火机制中温度的关系。从结果来看，混合算法在解的质量方面并没有大的突破，这可能与局部搜索部分没有涉及对提高解的质量的优化有关。因此未来的研究工作将进一步研究如何改善解的质量。并将通过大规模的实际应用问题来改善算法的求解性能。

# 第5章 总结与展望

随着全球经济一体化，国家贸易的更深入开展，国与国、区域与区域之间的各种货物往来会越来越多，物流行业的重要性越来越突出，而VRP问题作为物流行业调度环节的关键一环必须要解决好。在物流配送系统中，合理安排车辆路线是减少浪费、提高经济效益的重要手段，然而由于本身问题的NP完全性质，精确求解非常困难，研究启发式算法不失为一种可行的方向。通过对车辆路径问题的分析，本文首先给出了经典VRP问题的数学模型，现实生活中大多数VRP都可以由这个基本模型进行衍生。接着本文主要基于模拟退火算法的启发式算法求解了三种规模的VRP算例，并根据所得数据对算法进行了优化，将禁忌搜索与模拟退火法进行了混合。本文的创新之处有以下几点：一、在模拟退火算法中，通过数据分析发现是否有距离约束会在不同的温度下有不同的反应，所以对模拟退火进行了变温改进。二、通过采用更智能的迭代方式，并将模拟退火法与禁忌搜索进行混合，提高了搜索效率，在不降低解的质量的基础上大大减少了求解时间。本文的特色在于对算法使用C++程序的实现，并绘制出解的路径图，使其具有更强的实际意义。

目前，模拟退火算法己经在很多领域得到了应用，结合其他算法的混合算法更具有应用性，随着VRP问题的深入研究,相信很多问题都可以基于模拟退火的启发式算法来进行求解。

# 参考文献

[1] 宋燕子.[基于模拟退火法的启发式算法在VRP中的应用].武汉：华中师范大学，2013.

[2] O.Braysy and M.Gendreau.Vehiele routing Problem with time windows, Partl: Route construction and local search algorithms.Transportation Science, 39(l): 104-118, 2005.

[3] T.G.Grainic and G.Laporte, Planning models for freight transportation. European Journal of Operational Reasearch, 97:409-438, 1997.

[4] J.Larsen.Parallelization of the Vehicel Routing Problem with Time window. PhD thesis, Technical University of Denmark, 1999.

[5] G.Dantzig and J.Ramser. The truck dispatching problem. Management Science, 6:80-91, 1959.

[6] S.Ropke, J.-F.Cordeau, M.Iori and D.Vigo, “Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem with Two-Dimensional Loading Constraints.” Proceedings of ROUTE, Jekyll Island, 13 May 2007.

[7] A.G.Qureshi, E.Taniguchi and T.Yamada, “An Exact Solution Approach for Vehicle Routing and Scheduling Problems with Soft Time Windows,” Transprotation Research Part E: Logistics and Transportation Review, Vol.45, No.6, 2009, pp. 960-977. doi:10.1016/j.tre.2009.04.007.

[8] G.Gutiérrez-Jarpa, G.Desaulniers, G.Laporte and V.Ma- rianov, “A Branch-and-Price Algorithm for the Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows,” European Journal of Operational Re- search, Vol. 206, No.12, 2010, pp. 341-349.

[9] N.Azi, M.Gendreau and J.-Y.Potvin, “An Exact Algorithm for a Vehicle Routing Problem with Time Windows and Multiple Use of Vehicles,” European Journal of Operational Research, Vol.202, No.3, 2010, pp.756-763.

[10] Gilbert Laporte, Paolo Toth, Daniele Vigo, “Vehicle routing: historical perspective and recent contributions” EURO J Transp Logist (2013) 2:1–4.

[11] R. Montemanni, L. M. Gambardella, A. E. Rizzoli and A. V. Donati, “Ant Colony System for a Dynamic Vehicle Routing Pronlem,” Journal of Combinatorial Optimization, Vol. 10, No. 4, 2005, pp. 327-343.

[12] Y.-J. Cho and S.-D. Wang, “A Threshold Accepting Meta- Heuristic for the Vehicle Routing Problem with Backhauls and Time Windows,” Journal of the Eastern Asia Society for Transportation Studies, Vol. 6, 2005, pp. 3022-3037.

[13] M. Gandreau, F. Guertin, J.-Y. Potvin and R. Seguin, “Neighborhood Search Heuristics for a Dynamic Vehicle Dispatching Problem with Pick-Ups and Deliveries,” Transportation Research Part E: Logistics and Transportation Review, Vol. 14, No. 3, 2006, pp. 157-174.

[14] N. Yuichi and B. Olli, “A Powerful Route Minimization Heuristic for the Vehicle Routing Problem with Time Windows,” Operations Research Letters, Vol. 37, No. 5, 2009, pp. 333-338.

[15] G.-N. Abel and Bullinaria John A, “An Improved Multi-Objective Evolutionary Algorithm for the Vehicle Routing Problem with Time Windows,” Computers & Operations Research, Vol. 38, No. 1, 2011, pp. 287-300.

[16] K.-W. Peng, “An Adaptive Parallel Route Construction Heuristic for the Vehicle Routing Problem with Time Windows Constraints,” Expert System with Applications, Vol. 38, No. 9, 2011, pp.11939-11946.

[17] S. R. Balseiro, I. Loiseau and J. Ramone, “An Ant Col-ony Algorithm Hybridized with Insertion Heuristics for the Time Dependent Vehicle Routing Problem with Time Windows,” Computers & Operations Research, Vol.38, 2011, pp. 954-966.

[18] 肖朋等.车辆路径问题的单亲遗传算法[J].计算机技术与自动化,2000.

[19] 李军,谢秉晶,郭耀煌.非满载车辆调度问题的遗传算法[M].系统工程理论方法应用,2000,9 (3): 235-239

[20] 袁健,刘晋.随机需求情形VRP的Hopfield神经网络解法[J].南京航空航大大学学报,2000, 32 (5): 579-585.

[21] Zhu Chongjun, Liu Min’Wu Cheng. The effect of fuzzy demand possibility distribution and confidence level to the VRP solution. The First International Conference on Mechanical Engineering. ICME 2000.

[22] 刘浩,袁健,卢厚清.两种类型车辆随机需求路由问题[J].南京航空航天大学学报,2001,33 (2).

[23] 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——CIMS.2002, 8 (6): 451-454.

[24] 郎茂祥.用单亲遗传算法求解配送车辆调度问题的研究[J].交通与计算机,2006,128 (24): 119-122.

[25] 崔雪丽,马良,范炳全.车辆路径问题(VRP)的蚂蚁搜索算法[J].系统工程学报,2004,19 (4): 418-422.

[26] 刘云忠,宣慧玉.蚂蚁算法在车辆路径问题中的应用研宄[J].信息与控制,2004,33 (2): 249-252.

[27] 张建勇,李军.模糊车辆路径问题的一种混合遗传算法[J].管理工程学报,2005,19 (2): 23-26.

[28] 崔雪丽,朱振刚.VRP的混合改进型蚂蚁算法求解[J].苏州科技学院学报(工程技术版),2009,4(22): 62-66.

[29] Yu Bin and Yang Zhong Zhen, “An Ant Colony Optimization Model: The Period Vehicle Routing Problem with Time Windows,” Transportation Research Part E, Vol. 47, 2011, pp. 166-181.

[30] T.Stutzle. Local Search Algorithms for Combinatorial Problem – Analysis, Improvement New Applications. Infix, Augustin, 1999.

[31]

# 附录A 实验结果相关表格与图像

表A1 算例特征信息

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Dimension | Capacity | Distance | Name | Dimension | Capacity | Distance | Name | Dimension | Capacity | Distance |
| Christofied\_01 | 51 | 160 |  | Golden\_01 | 241 | 550 | 650 | Li\_21 | 561 | 1200 | 1800 |
| Christofied\_02 | 76 | 140 |  | Golden\_02 | 321 | 700 | 900 | Li\_22 | 601 | 900 | 1000 |
| Christofied\_03 | 101 | 200 |  | Golden\_03 | 401 | 900 | 1200 | Li\_23 | 641 | 1400 | 2200 |
| Christofied\_04 | 151 | 200 |  | Golden\_04 | 481 | 1000 | 1600 | Li\_24 | 721 | 1500 | 2400 |
| Christofied\_05 | 200 | 200 |  | Golden\_05 | 201 | 900 | 1800 | Li\_25 | 761 | 900 | 900 |
| Christofied\_06 | 51 | 160 | 200 | Golden\_06 | 281 | 900 | 1500 | Li\_26 | 801 | 1700 | 2500 |
| Christofied\_07 | 76 | 140 | 160 | Golden\_07 | 361 | 900 | 1300 | Li\_27 | 841 | 900 | 900 |
| Christofied\_08 | 101 | 200 | 230 | Golden\_08 | 441 | 900 | 1200 | Li\_28 | 881 | 1800 | 2800 |
| Christofied\_09 | 151 | 200 | 200 | Golden\_09 | 256 | 1000 |  | Li\_29 | 961 | 2000 | 3000 |
| Christofied\_10 | 200 | 200 | 200 | Golden\_10 | 324 | 1000 |  | Li\_30 | 687.9 | 2100 | 3200 |
| Christofied\_11 | 121 | 200 |  | Golden\_11 | 400 | 1000 |  | Li\_31 | 1121 | 2300 | 3500 |
| Christofied\_12 | 101 | 200 |  | Golden\_12 | 484 | 1000 |  | Li\_32 | 1201 | 2500 | 3600 |
| Christofied\_13 | 121 | 200 | 720 | Golden\_13 | 253 | 1000 |  |  |  |  |  |
| Christofied\_14 | 101 | 200 | 1040 | Golden\_14 | 321 | 1000 |  |  |  |  |  |
|  |  |  |  | Golden\_15 | 397 | 1000 |  |  |  |  |  |
|  |  |  |  | Golden\_16 | 481 | 1000 |  |  |  |  |  |
|  |  |  |  | Golden\_17 | 241 | 200 |  |  |  |  |  |
|  |  |  |  | Golden\_18 | 301 | 200 |  |  |  |  |  |
|  |  |  |  | Golden\_19 | 361 | 200 |  |  |  |  |  |
|  |  |  |  | Golden\_20 | 421 | 200 |  |  |  |  |  |

表A2 C1组变温实验

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Best\_1 | Best\_2 | Best\_3 | Best\_4 | Best\_5 | Best\_6 | Best\_7 | Best\_8 | Best\_9 | Best\_10 |
| Chris\_01 | 1.031 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Chris\_02 | 1.031 | 1.005 | 1.017 | 1.008 | 1.011 | 1.001 | 1.003 | 1.003 | 1.013 | 1.003 |
| Chris\_03 | 1.005 | 1.002 | 1.002 | 1.002 | 1.005 | 1.006 | 1.008 | 1.009 | 1.011 | 1.007 |
| Chris\_04 | 1.028 | 1.007 | 1.013 | 1.014 | 1.006 | 1.009 | 1.008 | 1.016 | 1.017 | 1.025 |
| Chris\_05 | 1.015 | 1.016 | 1.018 | 1.022 | 1.021 | 1.023 | 1.019 | 1.036 | 1.046 | 1.047 |
| Chris\_06 | 0.974 | 0.945 | 0.945 | 0.945 | 0.951 | 0.945 | 0.945 | 0.945 | 0.945 | 0.945 |
| Chris\_07 | 0.933 | 0.929 | 0.921 | 0.932 | 0.919 | 0.919 | 0.925 | 0.926 | 0.924 | 0.92 |
| Chris\_08 | 0.958 | 0.958 | 0.958 | 0.954 | 0.959 | 0.955 | 0.959 | 0.963 | 0.967 | 0.962 |
| Chris\_09 | 0.903 | 0.898 | 0.893 | 0.894 | 0.892 | 0.892 | 0.905 | 0.897 | 0.905 | 0.914 |
| Chris\_10 | 0.947 | 0.94 | 0.944 | 0.947 | 0.952 | 0.95 | 0.952 | 0.952 | 0.962 | 0.97 |
| Chris\_11 | 1.005 | 1 | 1 | 1.001 | 1.001 | 1.001 | 1.003 | 1.004 | 1.005 | 1.004 |
| Chris\_12 | 1 | 1 | 1 | 1 | 1.001 | 1 | 1 | 1 | 1.003 | 1 |
| Chris\_13 | 0.68 | 0.676 | 0.676 | 0.677 | 0.677 | 0.677 | 0.678 | 0.679 | 0.679 | 0.679 |
| Chris\_14 | 0.946 | 0.946 | 0.946 | 0.946 | 0.947 | 0.946 | 0.946 | 0.946 | 0.949 | 0.946 |
| AVG | 0.964 | 0.951 | 0.952 | 0.953 | 0.953 | 0.951 | 0.953 | 0.955 | 0.959 | 0.958 |

表A3 C2组变温实验

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Best\_1 | Best\_2 | Best\_3 | Best\_4 | Best\_5 | Best\_6 | Best\_7 | Best\_8 | Best\_9 | Best\_10 |
| Golden\_01 | 1.027 | 1.026 | 1.017 | 1.015 | 1.015 | 1.03 | 1.024 | 1.018 | 1.034 | 1.034 |
| Golden\_02 | 1.04 | 1.042 | 1.018 | 1.013 | 1.011 | 1.01 | 1.023 | 1.011 | 1.012 | 1.012 |
| Golden\_03 | 1.059 | 1.047 | 1.025 | 1.029 | 1.011 | 1.015 | 1.014 | 1.006 | 1.012 | 1.011 |
| Golden\_04 | 1.092 | 1.056 | 1.026 | 1.011 | 1.017 | 1.022 | 1.03 | 1.013 | 1.013 | 1.015 |
| Golden\_05 | 1.066 | 1.033 | 1.03 | 1.031 | 1.001 | 1.03 | 1.021 | 1.03 | 1 | 1.001 |
| Golden\_06 | 1.042 | 1.041 | 1.021 | 1.036 | 1.021 | 1.021 | 1.007 | 1.021 | 1.021 | 1.007 |
| Golden\_07 | 1.055 | 1.036 | 1.029 | 1.016 | 1.005 | 1.013 | 1.025 | 1.016 | 1.023 | 1.002 |
| Golden\_08 | 1.076 | 1.049 | 1.037 | 1.029 | 1.026 | 1.022 | 1.088 | 1.022 | 1.025 | 1.023 |
| Golden\_09 | 1.023 | 1.029 | 1.056 | 1.063 | 1.079 | 1.097 | 1.085 | 1.094 | 1.085 | 1.085 |
| Golden\_10 | 1.021 | 1.031 | 1.055 | 1.077 | 1.098 | 1.111 | 1.109 | 1.089 | 1.103 | 1.114 |
| Golden\_11 | 1.016 | 1.034 | 1.055 | 1.089 | 1.079 | 1.084 | 1.109 | 1.11 | 1.104 | 1.095 |
| Golden\_12 | 1.024 | 1.037 | 1.063 | 1.09 | 1.105 | 1.114 | 1.072 | 1.122 | 1.127 | 1.118 |
| Golden\_13 | 1.024 | 1.021 | 1.034 | 1.041 | 1.058 | 1.067 | 1.084 | 1.075 | 1.07 | 1.071 |
| Golden\_14 | 1.016 | 1.024 | 1.034 | 1.055 | 1.065 | 1.07 | 1.093 | 1.085 | 1.087 | 1.081 |
| Golden\_15 | 1.02 | 1.021 | 1.028 | 1.047 | 1.062 | 1.087 | 1.095 | 1.09 | 1.095 | 1.086 |
| Golden\_16 | 1.019 | 1.025 | 1.034 | 1.06 | 1.089 | 1.083 | 1.095 | 1.097 | 1.097 | 1.103 |
| Golden\_17 | 1.015 | 1.013 | 1.042 | 1.029 | 1.053 | 1.053 | 1.059 | 1.05 | 1.05 | 1.056 |
| Golden\_18 | 1.021 | 1.026 | 1.031 | 1.051 | 1.056 | 1.051 | 1.054 | 1.069 | 1.056 | 1.058 |
| Golden\_19 | 1.014 | 1.019 | 1.025 | 1.043 | 1.047 | 1.054 | 1.058 | 1.06 | 1.063 | 1.064 |
| Golden\_20 | 1.015 | 1.019 | 1.021 | 1.026 | 1.041 | 1.046 | 1.047 | 1.054 | 1.051 | 1.065 |
| AVG | 1.03425 | 1.03145 | 1.03405 | 1.04255 | 1.04695 | 1.054 | 1.0596 | 1.0566 | 1.0564 | 1.05505 |

表A4 C3组变温实验

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Best\_1 | Best\_2 | Best\_3 | Best\_4 | Best\_5 | Best\_6 | Best\_7 | Best\_8 | Best\_9 | Best\_10 |
| Li\_21 | 1.079 | 1.062 | 1.029 | 1.028 | 1.019 | 1.026 | 1.027 | 1.018 | 1.027 | 1.027 |
| Li\_22 | 1.034 | 1.023 | 1.025 | 1.026 | 1.023 | 1.025 | 1.024 | 1.024 | 1.027 | 1.027 |
| Li\_23 | 1.062 | 1.041 | 1.031 | 1.031 | 1.018 | 1.027 | 1.02 | 1.027 | 1.02 | 1.027 |
| Li\_24 | 1.072 | 1.04 | 1.015 | 1.003 | 1.014 | 1.01 | 1.013 | 1.028 | 1.028 | 1.007 |
| Li\_25 | 1.029 | 1.027 | 1.025 | 1.027 | 1.027 | 1.027 | 1.025 | 1.043 | 1.032 | 1.041 |
| Li\_26 | 1.112 | 1.079 | 1.053 | 1.028 | 1.012 | 1.011 | 1.011 | 1.006 | 1.007 | 1.012 |
| Li\_27 | 1.052 | 1.034 | 1.027 | 1.026 | 1.028 | 1.03 | 1.03 | 1.036 | 1.043 | 1.039 |
| Li\_28 | 1.087 | 1.074 | 1.034 | 1.03 | 1.03 | 1.042 | 1.031 | 1.01 | 1.027 | 1.031 |
| Li\_29 | 1.085 | 1.042 | 1.044 | 1.032 | 1.03 | 1.031 | 1.03 | 1.03 | 1.03 | 1.03 |
| Li\_30 | 1.123 | 1.09 | 1.099 | 1.076 | 1.035 | 1.052 | 1.031 | 1.031 | 1.031 | 1.03 |
| Li\_31 | 1.104 | 1.033 | 1.041 | 1.028 | 1.043 | 1.04 | 1.012 | 1.031 | 1.034 | 1.034 |
| Li\_32 | 1.097 | 1.063 | 1.056 | 1.046 | 1.048 | 1.021 | 1.062 | 1.043 | 1.042 | 1.045 |
| AVG | 1.078 | 1.050667 | 1.039917 | 1.03175 | 1.02725 | 1.0285 | 1.026333 | 1.02725 | 1.029 | 1.029167 |

表A6 C1组混合算法结果

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name/Tempra | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | AVG |
| Christofied\_01 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Time | 0.53 | 0.61 | 0.56 | 0.53 | 0.58 | 0.51 | 0.61 | 0.64 | 0.51 | 0.64 | 0.572 |
| Christofied\_02 | 1.01 | 1.012 | 1.013 | 1.014 | 1.016 | 1.009 | 1.004 | 1.01 | 1.001 | 1.013 | 1.0102 |
| Time | 1.22 | 1.2 | 1.31 | 1.28 | 1.42 | 1.03 | 1.16 | 0.94 | 1.37 | 1.26 | 1.219 |
| Christofied\_03 | 1.003 | 1.003 | 1.002 | 1.002 | 1.004 | 1.004 | 1.004 | 1.004 | 1.006 | 1.003 | 1.0035 |
| Time | 2.17 | 2.18 | 2.15 | 1.97 | 2.18 | 1.81 | 2.11 | 2.37 | 1.7 | 2.06 | 2.07 |
| Christofied\_04 | 1.012 | 1.012 | 1.014 | 1.008 | 1.015 | 1.01 | 1.016 | 1.017 | 1.007 | 1.012 | 1.0123 |
| Time | 3.28 | 3.84 | 3.84 | 4.1 | 3.77 | 3.65 | 3.4 | 3.81 | 4.01 | 3.71 | 3.741 |
| Christofied\_05 | 1.019 | 1.02 | 1.019 | 1.017 | 1.021 | 1.024 | 1.023 | 1.02 | 1.023 | 1.025 | 1.0211 |
| Time | 4.96 | 4.48 | 5.8 | 5.27 | 6.18 | 6.38 | 5.82 | 5.49 | 6.38 | 5.96 | 5.672 |
| Christofied\_06 | 0.945 | 0.945 | 0.945 | 0.945 | 0.945 | 0.945 | 0.945 | 0.945 | 0.945 | 0.945 | 0.945 |
| Time | 0.95 | 0.81 | 0.76 | 0.76 | 0.81 | 0.89 | 0.8 | 0.94 | 0.72 | 0.62 | 0.806 |
| Christofied\_07 | 0.919 | 0.92 | 0.933 | 0.929 | 0.919 | 0.935 | 0.927 | 0.924 | 0.934 | 0.919 | 0.9259 |
| Time | 1.81 | 1.78 | 1.64 | 1.7 | 1.9 | 1.65 | 1.9 | 1.73 | 1.37 | 1.84 | 1.732 |
| Christofied\_08 | 0.958 | 0.955 | 0.958 | 0.958 | 0.954 | 0.955 | 0.955 | 0.958 | 0.959 | 0.955 | 0.9565 |
| Time | 3.6 | 3.45 | 3.59 | 3.81 | 3.48 | 3.62 | 3.38 | 3.52 | 3.54 | 3.35 | 3.534 |
| Christofied\_09 | 0.896 | 0.894 | 0.897 | 0.892 | 0.895 | 0.897 | 0.892 | 0.893 | 0.894 | 0.891 | 0.8941 |
| Time | 5.43 | 5.54 | 6.08 | 5.19 | 4.84 | 5.21 | 5.72 | 5.1 | 5.12 | 5.29 | 5.352 |
| Christofied\_10 | 0.949 | 0.944 | 0.94 | 0.945 | 0.944 | 0.945 | 0.944 | 0.939 | 0.946 | 0.939 | 0.9435 |
| Time | 6.05 | 8.1 | 7.88 | 7.49 | 7.66 | 7.52 | 7.27 | 7.99 | 6.54 | 5.99 | 7.249 |
| Christofied\_11 | 1 | 1.004 | 1.001 | 1 | 1.001 | 1 | 1 | 1.004 | 1.001 | 1.001 | 1.0012 |
| Time | 2.54 | 2.81 | 3.04 | 2.9 | 3 | 3.23 | 2.78 | 2.85 | 3 | 2.82 | 2.897 |
| Christofied\_12 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Time | 2.26 | 1.95 | 2.35 | 2.4 | 2.11 | 2.03 | 1.97 | 2.37 | 1.87 | 2 | 2.131 |
| Christofied\_13 | 0.679 | 0.676 | 0.679 | 0.676 | 0.677 | 0.676 | 0.678 | 0.676 | 0.679 | 0.676 | 0.6772 |
| Time | 3.23 | 4.2 | 4.07 | 4.18 | 4.37 | 2.81 | 3.63 | 3.35 | 3.62 | 4.09 | 3.755 |
| Christofied\_14 | 0.946 | 0.946 | 0.946 | 0.946 | 0.946 | 0.946 | 0.946 | 0.946 | 0.946 | 0.946 | 0.946 |
| Time | 2.85 | 2.75 | 3.32 | 3.59 | 2.64 | 2.98 | 2.79 | 2.84 | 2.82 | 3 | 2.958 |

表A7 C2组混合算法

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | AVG |
| Golden\_01 | 1.028 | 1.03 | 1.015 | 1.027 | 1.018 | 1.033 | 1.025 | 1.028 | 1.022 | 1.027 | 1.0253 |
| Time | 7.44 | 7.19 | 6.55 | 6.46 | 7.91 | 7.13 | 6.1 | 6.63 | 8.02 | 5.57 | 6.9 |
| Golden\_02 | 1.019 | 1.013 | 1.013 | 1.017 | 1.014 | 1.03 | 1.016 | 1.017 | 1.031 | 1.022 | 1.0192 |
| Time | 10.44 | 10 | 10.53 | 10.69 | 12.67 | 13.12 | 10.53 | 10.27 | 9.89 | 10.44 | 10.858 |
| Golden\_03 | 1.013 | 1.026 | 1.027 | 1.015 | 1.022 | 1.029 | 1.03 | 1.02 | 1.024 | 1.032 | 1.0238 |
| Time | 15.96 | 15.15 | 16.96 | 12.84 | 16.82 | 18.77 | 16.4 | 16.6 | 13.35 | 15.76 | 15.861 |
| Golden\_04 | 1.015 | 1.011 | 1.014 | 1.017 | 1.012 | 1.011 | 1.015 | 1.008 | 1.007 | 1.012 | 1.0122 |
| Time | 20.51 | 20.17 | 21.18 | 20.62 | 18.94 | 19.75 | 19.59 | 19.53 | 18.78 | 21.25 | 20.032 |
| Golden\_05 | 1.031 | 1.031 | 1.002 | 1.029 | 1.03 | 1.031 | 1.032 | 1.031 | 1.031 | 1.031 | 1.0279 |
| Time | 4.17 | 4.07 | 3.35 | 4.07 | 3.23 | 2.93 | 3.38 | 3.88 | 4.07 | 4.04 | 3.719 |
| Golden\_06 | 1.021 | 1.022 | 1.021 | 1.007 | 1.021 | 1.007 | 1.021 | 1.021 | 1.021 | 1.021 | 1.0183 |
| Time | 7.29 | 7.55 | 7.63 | 8.06 | 9.28 | 7.72 | 6.43 | 7.05 | 7.33 | 5.77 | 7.411 |
| Golden\_07 | 1.021 | 1.019 | 1.027 | 1.007 | 1.015 | 1.016 | 1.019 | 1.025 | 1.021 | 1.023 | 1.0193 |
| Time | 12.62 | 11.26 | 13.26 | 11.36 | 11.95 | 11.78 | 12.48 | 12.32 | 11.65 | 10.81 | 11.949 |
| Golden\_08 | 1.032 | 1.026 | 1.038 | 1.028 | 1.038 | 1.031 | 1.034 | 1.036 | 1.03 | 1.026 | 1.0319 |
| Time | 15.21 | 17.99 | 18.72 | 17.6 | 17.29 | 15.01 | 17.49 | 18.38 | 14.38 | 17.83 | 16.99 |
| Golden\_09 | 1.021 | 1.029 | 1.026 | 1.024 | 1.02 | 1.027 | 1.022 | 1.023 | 1.022 | 1.024 | 1.0238 |
| Time | 7.19 | 5.54 | 5.45 | 7.18 | 5.69 | 5.19 | 6.96 | 5.46 | 5.83 | 7.02 | 6.151 |
| Golden\_10 | 1.023 | 1.024 | 1.023 | 1.029 | 1.02 | 1.023 | 1.024 | 1.02 | 1.022 | 1.022 | 1.023 |
| Time | 14.24 | 10.28 | 9.98 | 18.98 | 10.7 | 10.22 | 9.52 | 9.45 | 11.86 | 7.91 | 11.314 |
| Golden\_11 | 1.019 | 1.024 | 1.02 | 1.026 | 1.019 | 1.025 | 1.03 | 1.022 | 1.017 | 1.021 | 1.0223 |
| Time | 9.89 | 14.02 | 14.62 | 12.78 | 15.6 | 16.5 | 13.62 | 13.95 | 13.53 | 14.3 | 13.881 |
| Golden\_12 | 1.033 | 1.028 | 1.026 | 1.028 | 1.022 | 1.019 | 1.027 | 1.02 | 1.023 | 1.026 | 1.0252 |
| Time | 17.88 | 18.14 | 20.03 | 18.24 | 19.3 | 19.84 | 19.41 | 18.14 | 18 | 18.27 | 18.725 |
| Golden\_13 | 1.03 | 1.03 | 1.026 | 1.024 | 1.028 | 1.023 | 1.025 | 1.028 | 1.024 | 1.029 | 1.0267 |
| Time | 4.99 | 7.6 | 6.16 | 7.27 | 5.65 | 5.87 | 5.91 | 7.1 | 7.14 | 7 | 6.469 |
| Golden\_14 | 1.029 | 1.03 | 1.025 | 1.022 | 1.026 | 1.025 | 1.026 | 1.026 | 1.026 | 1.023 | 1.0258 |
| Time | 9.47 | 9.84 | 9.77 | 10.53 | 10.31 | 10.19 | 11.22 | 10.25 | 9.97 | 10.87 | 10.242 |
| Golden\_15 | 1.028 | 1.027 | 1.022 | 1.022 | 1.021 | 1.025 | 1.029 | 1.031 | 1.024 | 1.025 | 1.0254 |
| Time | 14.99 | 14.13 | 13.46 | 14.73 | 14.68 | 13.4 | 14.1 | 13.91 | 14.96 | 13.84 | 14.22 |
| Golden\_16 | 1.031 | 1.029 | 1.029 | 1.03 | 1.027 | 1.028 | 1.023 | 1.029 | 1.03 | 1.028 | 1.0284 |
| Time | 19.33 | 19.09 | 18.41 | 17.88 | 19.14 | 18.61 | 18.88 | 20.16 | 18.84 | 18.13 | 18.847 |
| Golden\_17 | 1.009 | 1.007 | 1.013 | 1.004 | 1.008 | 1.008 | 1.008 | 1.007 | 1.006 | 1.007 | 1.0077 |
| Time | 8.44 | 9.53 | 9.86 | 8.69 | 8.53 | 9.27 | 9.88 | 8.78 | 8.91 | 9.17 | 9.106 |
| Golden\_18 | 1.019 | 1.026 | 1.021 | 1.018 | 1.021 | 1.02 | 1.021 | 1.022 | 1.018 | 1.021 | 1.0207 |
| Time | 12.54 | 12.95 | 11.76 | 12.12 | 11.82 | 11.84 | 11.9 | 11.75 | 12.25 | 11.17 | 12.01 |
| Golden\_19 | 1.02 | 1.021 | 1.019 | 1.018 | 1.021 | 1.018 | 1.017 | 1.02 | 1.021 | 1.017 | 1.0192 |
| Time | 15.04 | 15.26 | 14.87 | 15.93 | 15.46 | 14.65 | 16.89 | 16.93 | 16.71 | 14.87 | 15.661 |
| Golden\_20 | 1.018 | 1.021 | 1.019 | 1.016 | 1.021 | 1.023 | 1.016 | 1.018 | 1.018 | 1.019 | 1.0189 |
| Time | 19.73 | 19.59 | 18.19 | 20.09 | 20.44 | 19.59 | 19.91 | 20.47 | 20.12 | 19.52 | 19.765 |

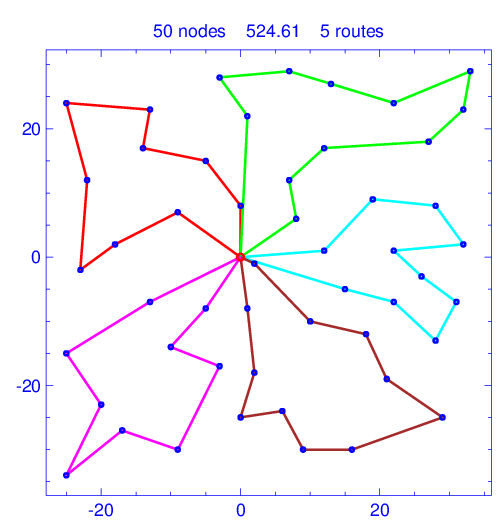
表A8 C3组混合算法

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | AVG |
| Li\_21 | 1.033 | 1.028 | 1.028 | 1.028 | 1.03 | 1.031 | 1.029 | 1.028 | 1.03 | 1.023 | 1.0288 |
| Time | 24.01 | 23.98 | 20.25 | 25.43 | 22.89 | 24.88 | 24.38 | 24.35 | 23.41 | 22.07 | 23.565 |
| Li\_22 | 1.028 | 1.028 | 1.026 | 1.028 | 1.03 | 1.027 | 1.024 | 1.025 | 1.027 | 1.027 | 1.027 |
| Time | 27.88 | 29.08 | 28.78 | 28.03 | 30.73 | 28.92 | 28.67 | 30.08 | 28.75 | 28.52 | 28.944 |
| Li\_23 | 1.023 | 1.027 | 1.017 | 1.021 | 1.021 | 1.027 | 1.018 | 1.022 | 1.013 | 1.018 | 1.0207 |
| Time | 26.79 | 28.88 | 27.05 | 27.11 | 32.26 | 29.38 | 28.56 | 25.63 | 30.4 | 27.75 | 28.381 |
| Li\_24 | 1.013 | 1.028 | 1.028 | 1.028 | 1.028 | 1.007 | 1.006 | 1.013 | 1.013 | 1.022 | 1.0186 |
| Time | 33.06 | 30.53 | 34.21 | 28.84 | 36.4 | 36.88 | 36.33 | 37.07 | 33.23 | 33.06 | 33.961 |
| Li\_25 | 1.027 | 1.025 | 1.028 | 1.027 | 1.029 | 1.025 | 1.026 | 1.028 | 1.025 | 1.027 | 1.0267 |
| Time | 43.62 | 38.49 | 43.12 | 42.54 | 46.75 | 43.91 | 44.46 | 48.72 | 42.31 | 42.46 | 43.638 |
| Li\_26 | 1.011 | 1.011 | 1.011 | 1.012 | 1.016 | 1.011 | 1.012 | 1.017 | 1.013 | 1.011 | 1.0125 |
| Time | 36.77 | 33.12 | 30.81 | 42.2 | 44.38 | 35.37 | 32.56 | 38.98 | 34.93 | 39.67 | 36.879 |
| Li\_27 | 1.033 | 1.033 | 1.028 | 1.032 | 1.034 | 1.034 | 1.044 | 1.029 | 1.043 | 1.031 | 1.0341 |
| Time | 51.85 | 61.37 | 56.6 | 55.47 | 53.52 | 54.3 | 55.54 | 56.78 | 49.22 | 56.91 | 55.156 |
| Li\_28 | 1.029 | 1.03 | 1.034 | 1.034 | 1.029 | 1.034 | 1.035 | 1.029 | 1.034 | 1.035 | 1.0323 |
| Time | 46.61 | 42.46 | 46.71 | 37.5 | 43.63 | 38.58 | 44.57 | 53.49 | 40.53 | 40.54 | 43.462 |
| Li\_29 | 1.009 | 1.009 | 1.009 | 1.009 | 1.014 | 1.009 | 1.009 | 1.009 | 1.004 | 1.014 | 1.0095 |
| Time | 48.58 | 56.18 | 48.41 | 46.18 | 57.99 | 48.66 | 61.01 | 57.35 | 47.56 | 52.92 | 52.484 |
| Li\_30 | 1.013 | 1.014 | 1.013 | 1.013 | 1.013 | 1.008 | 1.013 | 1.009 | 1.008 | 1.013 | 1.0117 |
| Time | 56.8 | 55.02 | 61.25 | 62.91 | 57.92 | 56.21 | 51.73 | 59.83 | 58.23 | 61.98 | 58.188 |
| Li\_31 | 1.037 | 1.033 | 1.038 | 1.035 | 1.036 | 1.033 | 1.037 | 1.033 | 1.037 | 1.033 | 1.0352 |
| Time | 78.69 | 68.05 | 69.34 | 64.05 | 64.35 | 74.85 | 73.34 | 64.23 | 65.66 | 72.53 | 69.509 |
| Li\_32 | 1.039 | 1.043 | 1.04 | 1.041 | 1.038 | 1.04 | 1.039 | 1.045 | 1.043 | 1.041 | 1.0409 |
| Time | 71.96 | 65.61 | 70.28 | 71.95 | 70.59 | 71.81 | 70.11 | 78 | 71.7 | 75.74 | 71.775 |

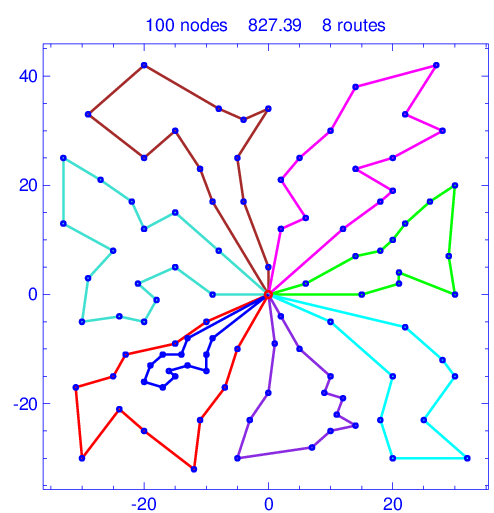
图A1 C1组变温实验

图A2 C2组变温实验

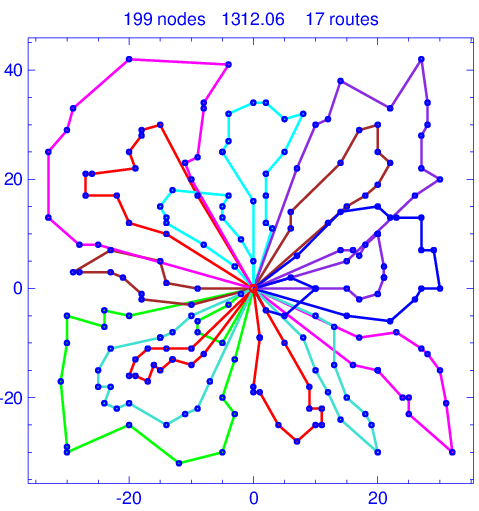
图A3 C3组变温实验



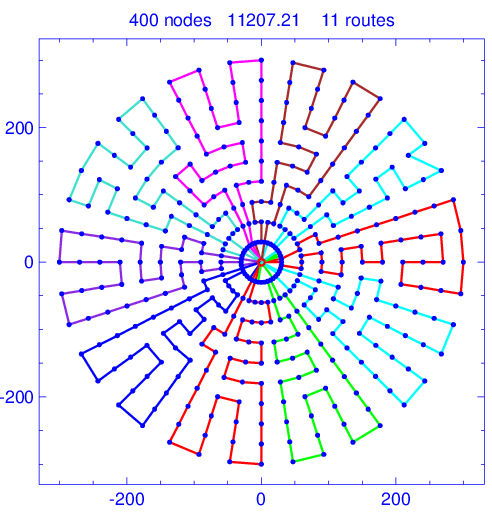
图A5 50个点路径图



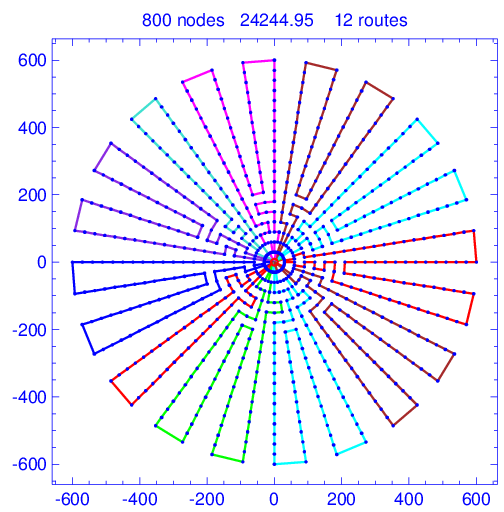
图A6 100个点路径图



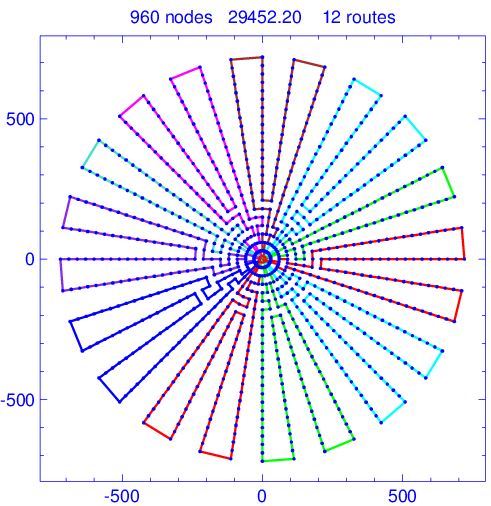
图A7 200个点路径图



图A8 400个点路径图



图A9 800个点路径图



图A10 960个点路径图

# 致 谢（黑体小二，居中）

毕业论文即将完成，我的学生生涯也要告一段落了。借此机会，我要对四年来帮助过我的人表示深深的感谢。

本文是在\*\*\*\*\*\*老师精心指导和大力支持下完成的，**………….（后面内容要补充完整）**

××××××××××××××××××××××××××××××××××××××××。

另外，我还要特别感谢同学对我论文研究分析以及论文写作的指导，为我完成这篇论文提供了很大的帮助。

**（宋体小四号）**

作者姓名

年 月