

上海交通大学

硕士学位论文

路径优化算法在集装箱堆场运营管理中的应用

姓名：刘义苍

申请学位级别：硕士

专业：管理科学与工程

指导教师：孙小明

20050201

路径优化算法在集装箱堆场运营管理中的应用

摘 要

作为一种新的服务和第三方利润源泉的现代物流业日益受到了广泛的关注,其中,车辆调度问题能否较好解决,直接关系到企业的经济效益,并且影响了我国整个物流产业的健康迅速发展。因此,车辆线路安排问题(Vehicle Routing Problem, 简记 VRP)成为了众多物流企业竞相研究应用的热门话题。为了适应现代物流的发展要求,作为集装箱运输链的重要环节——集装箱堆场需要按照现代物流的要求,合理配置内部物流要素,优化物流要素的流通渠道,完善物流信息系统,以不断提高堆场内部运作效率并降低运营成本。本文针对集装箱堆场运营管理中的瓶颈环节——搬运机调度安排问题进行了研究。首先针对该问题建立了数学模型,考虑了最小化搬运机总行驶路程和最小化集装箱拖车总延误时间两个目标,并将该多目标旅行商问题(Multi-objective Traveling Salesman Problem, 简记 MTSP)转化为含有时间惩罚函数的单目标成本问题。在优化算法上选择了模拟退火算法,并且对该问题的特殊情况——搬运机作业完所有点并不返回起始点的问题进行了探讨,得出了选择起始点为作业最终点的方法为较优方法。本文的研究成果已经成功运用于深圳市鸿基物流集装箱堆场有限公司的运营管理实践中,成功解决了该企业所面临的搬运机作业效率低下和集装箱拖车排队现象严重的现状。

关键词: VRP, TSP, 模拟退火算法, 集装箱堆场

Routing Optimization Algorithm used in Container

Yard Operation

Abstract

As a kind of new service and the third margin source, modern logistics is getting more attention. The planning of vehicle routing in third part logistics will make great effect on the efficiency, benefit and the healthy development of national logistics industry. Therefore, vehicle routing problem had become focus of many logistics corporation to study. As an important node in container transportation chain, container yard need configure internal logistics element in reason and optimize the circulating channel of logistics element so that boost the operation efficiency in container yard and decrease the cost. This paper studies the container crane routing problem which is one of the bottlenecks of operation management in container yard. A mathematic model with minimizing crane running distance and container truck waiting time is built in this paper. And the multi-object traveling salesman problem is translated into a single-object cost problem with time penalty function. Simulated Annealing algorithm is used to settle this problem. The problem which crane need not return the starting point after finishing all operation points is studied in this paper. And finally the method which selects starting point as the last operation point is certified as the best solution to this problem. The research result has increased the crane

operation efficiency and decrease the container truck waiting time with low cost when it was applied in Shenzhen Hongkai Logistics Container Yard.

Key word: VRP, TSP, Simulated Annealing algorithm, Container Yard

FEBRUARY 2005

上海交通大学 学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：刘义苍

日期：2005 年 2 月 18 日

上海交通大学 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留、使用学位论文的规定，同意学校保留并向国家有关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。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，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。

保密 ☐，在____年解密后适用本授权书。

本学位论文属于

不保密 ☐。

（请在以上方框内打“√”）

学位论文作者签名：刘义苍

指导教师签名：孙小明

日期：2005 年 2 月 18 日

日期：2005 年 2 月 18 日

上海交通大学学位论文答辩决议书

申 请 者	刘义苍	所在学科 (专业)	管理科学与工程
论 文 题 目	路径优化算法在集装箱堆场运营管理中的作用		
答 辩 日 期	2005 年 2 月 21 日	地 点	机械楼 303 室
答 辩 委 员 会 成 员			
姓 名	单 位	职 称	签 名
胡宗武	上海交通大学工业工程与管理系	教授、博导	胡宗武
江志斌	上海交通大学工业工程与管理系	教授、博导	江志斌
王丽亚	上海交通大学工业工程与管理系	教授	王丽亚
孙小明	上海交通大学工业工程与管理系	副教授	孙小明
苏 强	上海交通大学工业工程与管理系	副教授	苏 强

评语和决议:

运输路线优化问题的研究对于提高企业物流运作效率具有重要意义。

论文应用模拟退火算法对深圳市鸿基物流堆场有限公司集装箱堆场车辆路线安排问题进行优化求解,使原先先到先服务的方法行程有很大缩短,从而使集装箱拖车总等待时间从 182 分钟缩减到 153 分钟,具有较大的经济效益和实用价值。

论文理论分析严密正确,文字描述流畅,可读性强;从论文看,作者研究方法正确,具有较强的独立进行科学研究的能力。论文达到了硕士学位论文要求。在论文答辩中,表述清楚,回答问题正确。答辩委员会一致同意通过硕士论文答辩,并建议授予管理学硕士学位。

表决结果:

一致同意,建议授予硕士学位

答辩委员会主席 胡宗武 (签名)

2005 年 2 月 21 日

第一章 概述

1.1 课题背景与来源

1.1.1 现代物流产业中的车辆路径调度问题

20 世纪六七十年代, 发达国家的企业大都把竞争焦点放在生产领域, 千方百计地降低物质资源消耗, 获取企业的“第一利润源泉”, 以及千方百计提高劳动生产率获取企业的“第二利润源泉”。进入 20 世纪 70 年代以后, 随着科学技术的进步, “第一利润源”和“第二利润源”的空间越来越小。早在 1980 年的全美物资讨论会上, 研究者们就指出, 在产品的整个生产过程中, 仅仅 5%的时间用于加工和制造, 剩余的 95%都用于储存、装卸、等待加工和输送^{[1][2]}。直接劳动成本所占比例不足工厂总成本的 10%, 并且这一比例还在不断下降, 而储存、运输所支付的费用却占生产成本的 40%。因此, 生产企业把提高经济效益、增强竞争力的焦点从生产领域开始转向非生产领域, 包括采购、运输、储存、包装、装卸、流通加工、分销、售后服务等物资流通活动以及有关的信息活动, 即通过整合企业现代物流活动以获得“第三利润源”。^{[1][3]}

现代物流的基本涵义为“按用户要求, 将物的实体从供应地到需求地转移的过程”(Martha and Cooper, 2001)。物流产业涉及铁路运输、公路运输、内河和海上运输、管道运输、航空运输、邮政业、仓储业、装卸业、包装业、配送业、流通加工业、物流信息业等。可以发现, 大部分物流产业都可以归类为车辆路径调度问题, 例如除了铁路、公路等运输业, 装卸、配送等也可以归类为车辆路径调度问题的有机组成部分。目前, 我国物流产业效率较低, 损失浪费十分惊人。有专家估计, 我国仅汽车空驶率就达 37%, 相当于 150 万辆载重汽车来回空跑(中国物流与采购联合会, 2002)^[4]。由此可见, 车辆路径调度问题能否较好解决, 直接关系到企业的经济效益, 影响我国整个物流产业的健康迅速发展。

在竞争日益激烈的今天。企业只有以市场为核心去适应不断变化的环境并及时对市场做出反应, 以低的成本、快的速度, 在正确的时间和地点为消费者或用户提供满意的产品和服务, 才能在竞争中占有优势。产品分销渠道的优化是达到上述目标的关键步骤之一。分销渠道是指产品在其所有权转移过程中从生产领域进入消费领域的途径, 而分销渠道中的物流管理则是指通过有效地安排产品的仓储、运输、管理和转移, 使产品在需要的时间到达需要的地点的经营活动。在物流管理决策中, 车辆路径问题(Vehicle Routing Problems, 简称 VRP)是得到较多关注的一个方面。

1.1.2 课题的背景与研究的目标

本课题来源于“深圳市鸿基物流集装箱堆场管理信息系统开发项目”, 课题的目标是通过信息化技术的应用, 实现客户满意率的提高以及作业成本的降低,

结合深圳市鸿基物流集装箱堆场公司的运作实际状况,就是顾客来存或取集装箱的排队等待时间很长,最长的排队可以达到近 10 公里,另外,就是存、取集装箱的搬运设备的油耗过高,造成运作成本过高。

在现代物流产业链中,集装箱运输业由于其诸多优越性如:提高装卸效率,减轻劳动强度;加速车船周转,加快货物送达;便于实现“门到门”运输;避免货物捣载,防止货损货差;节约包装费用,简化管理手续;便于采用计算机管理,实现标准化作业等,使得集装箱运输在现代物流中日益占据了举足轻重的地位。随着现代科学技术的不断进步和日益激烈的国际市场竞争,集装箱运输已经由传统的“货运业”向现代化的“物流业”发展,从过去我们所熟知的海上运输向内陆多式联运范围不断延伸。

作为集装箱运输的一个重要环节——大型集装箱堆场在这里更是显示了极大的优势,它承载了大宗货物的存储和运输,已经成为集装箱运输不可缺少的一个重要环节。堆场提供的服务已经深深地参与到了集装箱运输的生产活动中去,实实在在地为集装箱货主提供了增值服务,例如随着集装箱运输通过铁路、公路等运输模式,在内陆地区也相应出现了许多集装箱堆场,当然其名称根据其服务特色,分别称为集装箱中转站,集装箱公路港,集装箱铁路专用车站等等,所有这些规模范围大大小小的内陆集装箱堆场都是远离港口码头,服务于内陆腹地,为集装箱经营人开拓内陆市场,提高集装箱运输的经济效益做出巨大的贡献。

仓储、堆场发展历史比较长远,但是仓储、堆场实现自动化管理的历史并不长。从其整个发展过程来看,可以分为五个阶段^[5]:

(1)人工储运阶段。在仓储、堆场出现的初期,由于科学不发达,物资的运输、储存、管理和控制基本上是靠人工实现的。人工储运耗费大量劳力,并且速度和效率很低,满足不了生产的要求。但是人工储运具有较高的实时性和灵活性。

(2)机械化储运阶段。随着科学技术的发展,各种传输、搬运机械引入到仓储和堆场中。物资的传送运输可以通过传输带、吊车、堆垛机以及装卸桥等,储存物资可以通过货架、集装箱、托盘以及旋转托架等,各种电器装置实现储运、运输设备间的相互控制。

(3)自动化储运阶段。各种控制装置的出现,以及将计算机技术引入到仓储、堆场中,从而使仓储进入到自动化发展阶段。五十年代末和六十年代初,相继研制和采用了自动托盘和自动导引小车。七十年代以后,机器人以及各种自动存取装置应用到仓储中,使仓储中的存、运机械装置实现自动化作业;随着计算机技术的不断发展,仓储的管理和机械设备的控制都进入自动化系统管理阶段。

计算机技术的发展,也使仓储工作的重点转向了要求物资控制和管理达到实时、协调和一体化。计算机之间、数据采集设备之间、机械设备的控制器之间以及它们和主计算机之间的通信都可以实时进行。在自动化管理系统的监控下,仓储的效率大大提高。

(4)集成化储运阶段。它强调中央控制下各个自动化物流设备的统一和协同操作,并且各物流设备的指挥统一由主计算机来完成。从而对各个设备间的工作状况、位置信息等实时情况进行有效地监控,并且通过控制中心实现各个设备间的协调。

(5)智能化储运阶段。将人工智能的有关理论、特别是专家系统的相关知识应用于控制计算机,使其能够智能化地处理相关问题。在物流系统中,控制中心结合物料的需求以及存储情况,根据专家系统做出各种出库、入库决策,确保运输、生产的有效进行。

集装箱堆场在我国的发展起步比较晚,但是发展速度相当快,而且随着以通讯技术、网络技术、感测技术、控制技术为代表的现代信息技术的快速发展以及全球信息网络的兴起,现在已经初具规模。虽然日前我国一些集装箱堆场已具备了先进的生产设施和现场实时控制系统,但总体来说,与国外先进的集装箱堆场相比还存在一定差距。主要表现在对堆场内部物流系统的资源配置、任务安排和路径优化等方面还采用经验管理,堆场物流信息系统也有待于进一步完善和发展。为了适应现代物流和供应链管理的发展,满足快速增加的箱量需求和提高顾客服务水平,在新一轮的竞争中取得优势,堆场管理者需按照现代物流和供应链管理的要求,不断提高堆场内部物流网络节点配置的合理性、充分重视集装箱装卸时机械设备作业计划和作业顺序制定的科学性、并不断完善其内部物流信息系统,以提高资源利用率、提高作业效率和效益。同时物流技术的发展和广泛应用也为集装箱堆场提高其内部物流运作效率提供了保障。

在集装箱堆场管理中,搬运机的作业调度直接反应了堆场作业成本的高低,而等候装卸作业的集装箱拖车在堆场中的滞留时间是反应堆场客户服务水平的一个重要指标。目前,大多数堆场的作业成本一直也无法得到有效的降低,搬运机的油耗成本和维护成本都偏高。而且由于搬运机的作业效率低下等问题导致了大多数堆场都不同程度的出现了在高峰作业阶段等待作业的集装箱拖车排成长龙,一方面影响公司的服务形象,降低了市场的竞争能力,另一方面,严重阻碍了公共交通的情况。实际操作中,现场管理人员也想出了各种办法来解决这个问题,如:提高进出场检验、登记的自动化水平,增加堆场作业设备数量或购买作业效率更高的作业设备,以及加强信息化管理水平和改善操作流程等等。本文将作业设备调度研究为主要目标,运用运筹优化方法优化作业设备的作业行驶路径,提高作业设备的作业效率,降低作业设备的空驶率,同时减少集装箱拖车在场等待时间,提高顾客服务水平,希望能为集装箱堆场内部物流网络运作提供一定的帮助。

1.2 论文的研究内容

针对目前大多数集装箱堆场在实际操作中采用先到先服务的原则进行进出货装卸操作,本文提出运用现代优化算法——模拟退火算法对堆场上等待作业的集装箱拖车的作业次序进行重新排序,以达到减少集装箱拖车在场等待时间,降低作业设备的空驶率,从而提高客户服务水平和作业效率。本文首先建立作业设备调度问题的数学模型,以作业设备总行驶路程和各集装箱拖车在场等待时间为目标,研究多目标车辆路线安排问题(MVRP),即实现最小化搬运机总行驶路程和最小化集装箱拖车总延误时间两个目标。由于MVRP问题不同目标之间相互制约,使其建模和求解具有一定的难度,文中引入一个排序变量 z_{ix} ,即当第 i 个集装箱操作排在第 x 位置时 $z_{ix}=1$;否则, $z_{ix}=0$,并且分别引入平均单位运输成本 C_1 和平均单位等待时间惩罚成本 C_2 将作业设备总行驶路程和各集装箱拖车在场总等待时间分别转化为总运输成本和总时间惩罚成本,从而将作业设备总行驶路程目标和各集装箱拖车在场总等待时间目标统一转化为总成本目标,实现了将原多目标问题转化成单目标问题求解。

文章首先在第二章中讨论了路径优化问题如 VRP 问题和 TSP 问题的研究现状, 并且讨论了目前常用的各种路径优化算法的研究现状以及优缺点, 最终选择了模拟退火算法作为该论题的解决方法。在第三章中针对集装箱堆场的具体问题建立了数学模拟, 并且介绍了模拟退火算法和其中的各种参数选择方法。在第四章中确定了模拟退火算法的参数设计, 并且针对该问题的特殊情况: 作业始点和终点不为同一点的问题进行了探讨, 提出了三种方法, 分别为: 选起点为最终点的算法、选距离起点最远的点为最终点的算法和选距离起点最近的点为最终点的算法, 并且通过仿真实验得出选起点为最终点的方法为该问题的最优解决方法。然后就目前大多数堆场所常用的先到先服务的调度原则进行了对比, 得出了应用模拟退火算法优化了搬运机的作业路线, 降低了搬运机的空驶率, 提高了堆场作业效率, 使得堆场运营成本得到了大大的降低, 而且服务水平也得到了一定的提高。

1.3 论文的研究意义

随着我国经济的高速增长, 以及电子商务的蓬勃发展和中国加入 WTO, 当前物流活动呈现出前所未有的频繁, 物流业已成为我国国民经济新的增长点。但是目前我国物流管理仍然比较落后, 物流行业普遍面临着专业化程度低、高耗低效等问题。目前我国大多数的物流企业的车辆调度依然依赖于人工经验并采用人工安排的方式, 从而导致企业运输资源无法充分利用, 运营成本过高, 或者无法满足客户的要求。因此, 对车辆调度的优化问题进行研究, 建立能够即时满足客户需求的车辆调度系统, 是提升服务及资源利用率的重要课题。VRP 和 TSP 的研究作为发展敏捷后勤的一个重要组成部分, 是实现物流现代化的基础和前提条件, 不仅可以帮助运输企业提高服务水平, 为顾客提供快捷、准时、安全、舒适的服务, 而且有助于企业节约运输成本, 改善车辆利用效率, 缩短生产周期, 加速资金周转, 实现资源的合理配置, 汲取“第三利润源泉”的财富。本文对 VRP 和 TSP 问题的国内外研究成果作了简要介绍, 并且将其运用于深圳市鸿基物流集装箱堆场有限公司的运营管理实践中, 成功解决了该企业所面临的搬运机作业效率低下和集装箱拖车排队现象严重的现状。

第二章 车辆路径优化问题的研究现状综述

2.1 车辆路由问题概述

车辆路由问题 (Vehicle Routing Problem, VRP)^[6]最早由 Dantzig 于 1959 年提出,是运筹学中的热点研究问题,其原型是旅行商问题 (Traveling Salesman Problem, TSP)^[7]。郭耀煌、李军在《车辆优化调度》一书中将 VRP 定义为,对一系列发货点和(或)收货点,组成适当的行车路径,使车辆有序地通过它们,在满足一定约束条件的情况下,达到一定的目标^[8]。在许多场合,人们也将 VRP 称为 VSP (Vehicle Scheduling Problems),但是严格地讲,这两类问题是有区别的,通常 VRP 是指与空间有关的路径安排问题,而 VSP 是指与时间有关的车辆配置问题。本论文遵照大多数人的习惯,对两类问题不做严格区分,仍统称为 VRP。

车辆优化调度和路径优化问题最早是由 Dantzig 和 Ramser 于 1959 年提出的,由于这一问题的理论涉及多学科,很多时间问题的理论抽象都可归结为这一类问题,应用前景广阔。所以,很快便引起运筹学、应用数学、图论与网络分析、物流科学、交通运输工程、管理科学与工程、计算机应用等学科的专家、工程技术人员和管理者的极大重视,自此,一直成为运筹学与组合优化领域的前沿与研究热点问题。各学科专家对该问题进行了大量的理论研究及实验分析,取得了很大的进展。目前,该问题的形式已有很大发展,该问题已不仅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。如其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。

为了便于研究分析,该问题提出后,Linus(1981),Bodin(1983),Assad(1988)等众多学者从不同角度,按不同标准进行了分类。

1. 按研究目标可以分为:
 - 1) 最小化总运输成本;
 - 2) 最小化空驶里程;
 - 3) 最小化顾客等待时间;
 - 4) 最大化服务率;
 - 5) 其它目标,即以上几类目标的组合。
2. 根据问题的空间特性和时间特性的相对重要性可以分为:
 - 1) 旅行商问题 (Traveling Salesman Problem, TSP),即不考虑时间要求和空间位置安排;
 - 2) 车辆线路安排问题 (Vehicle Routing Problem, VRP),即不考虑时间要求,仅根据空间位置安排线路;
 - 3) 称为车辆调度问题 (Vehicle Scheduling Problem, VSP),即仅考虑时间要求安排线路;
 - 4) Routing 和 Scheduling 混合问题 (Vehicle Routing and Scheduling Problem, VRP&VSP),即同时考虑空间位置和时间要求。
3. 按任务特征可以分为:
 - 1) 纯装问题或纯卸问题 (pure pick up or pure delivery),即车辆在

- 所有任务点装货或卸货，即集货或送货问题
- 2) 装卸混合问题(combined pick up and delivery)，即每项任务有不同的装货点和卸货点，即集货、送货一体化问题。
4. 按任务性质可以分为：
 - 1) 对弧服务问题(如中国邮递员问题)；
 - 2) 对点服务问题(如旅行商问题)；
 - 3) 混合服务问题(如交通车线路安排问题)。
 5. 按车辆载货状况可以分为：
 - 1) 满载问题，即货运量不小于车辆容量，完成一项任务需要不止一辆车；
 - 2) 非满载问题，即货运量小于车辆容量，多项任务用一辆车。
 6. 按车场(或货场、配送中心等)数目可以分为：
 - 1) 单车场问题；
 - 2) 多车场问题。
 7. 按车辆类型数可以分为：
 - 1) 有单车型问题，即所有车辆容量相同；
 - 2) 多车型问题，即执行任务的车辆的容量不全相同。
 8. 按车辆对车场的所属关系可以分为：
 - 1) 车辆开放问题，即车辆可以不返回其发出车场；
 - 2) 车辆封闭问题，即车辆必须返回其发出车场。

2.2 VRP 问题国内外研究综述

经过四十多年的研究，VRP 领域取得了相当丰富的成果，为了把该问题的研究现状清晰地呈现在众人面前，此处对该问题的研究成果作以回顾：2.2.1 节回顾国内研究者对 VRP 的研究和探索；2.2.2 节总结国外学者对 VRP 研究的成果。

2.2.1 国内 VRP 的研究现状

根据所能掌握的资料，国内最先对 VRP 进行系统研究的是郭耀煌教授，他及其学生从 1989 年起对多车场、多车型等类型的问题进行了研究^[9]，并出版了国内 VRP 研究领域的第一部专著《车辆优化调度》^[8]。近年来，随着“物流热”的升温，国内学者对 VRP 的研究也开始重视起来。在国内研究者撰写的四篇综述文章中，汪寿阳等对定址——路径问题的研究进行了分析评述^[10]，祝崇隽等较全面地回顾了 VRP 领域的最新进展^[11]，袁庆达等总结了库存——路径问题研究的一些成果^[12]，郭耀煌等总结了动态 VRP 的特征、分类以及算法研究^[13]。此外大多数文献都是对 VRP 的模型和算法进行研究^[14-26]。文献[16, 17, 23, 24, 26]采用遗传算法求解了有/无时间窗的 VRP，文献[25]用禁忌搜索算法求解了软时间窗的多车型 VRP，文献[18-22]设计了序列优化启发式算法和表上改进法，并用于求解有时间窗的多车场、多车型问题。令人感兴趣的是，文献[14]将一个转炉计划问题看成有时间窗的 VRP，然后用遗传算法求解对应的旅行商问题，再用 Davis 编码对旅行商回路进行分解。文献[15]研究了模糊需求的路径问题，提出了基于概率的 2-opt 算法和基于需求上界的 2-opt 算法。

尽管国内学者对 VRP 的研究已有一些成果,但总的来说,研究还处于起步阶段,主要表现在以下几点:(1)研究者人数少,而且属于不同的单位和专业,造成彼此缺乏交流,甚至一些术语的译法尚未统一;(2)所研究的问题大多属于确定性问题,尤其集中于有/无时间窗的 VRP,随机 VRP 和其它复杂的问题尚未涉及;(3)研究多是侧重于学术研究,与实际应用尚有距离。

国内的一些研究者从实用开发角度出发对 VRP 也进行了研究,其中文献[27, 28]提出了以 GIS 为平台通过集成 VRP 算法开发车辆路径系统的设计思想。目前已经投放市场的路径软件有:武汉测绘科技大学奥发公司的“商业送配货地理信息系统”和北大方正的“路径规划系统”,它们只能满足用户最基本的需要,当出现不确定性信息或问题约束较为复杂时,这些软件就显得无能为力了。

2.2.2 国外 VRP 的研究现状

一、确定性 VRP 的研究现状

由于 VRP 的最初形式就是确定性的,并且求解相对容易,因此,到目前为止,研究仍主要集中于确定性问题。针对各种各样的确定性问题,人们提出了许多算法,包括精确算法和启发式算法,下面分别以这两大类算法为线索,回顾用于求解确定性 VRP 的主要算法。

到目前为止,已提出的精确算法种类较多,但从其本质特征来看,大致可以归为两类:有向树搜索算法、动态规划算法。

Laporte 等利用 VRP 与多重旅行商之间的关系,将 VRP 变形为旅行商问题后,再用分枝定界法求得问题最优解^[29]。

求解确定性 VRP 的动态规划算法最早由 Eilon 等提出^[31]。令 $c(S)$ 表示通过车场 0 和顾客子集 $S \subseteq V \setminus \{0\}$ 中所有顾客的最小成本, $f_k(U)$ 为使用 k 辆车将货物运至顾客子集 $U \subseteq V \setminus \{0\}$ 的最小成本,可由下面的递推公式确定:

$$f_k(U) = \begin{cases} c(U), k=1 \\ \min_{U^* \subset U \subset V \setminus \{0\}} [f_{k-1}(U \setminus U^*) + c(U^*)], k > 1 \end{cases} \quad (2.1)$$

式 2.1 最终求得的成本为 $f_m(V \setminus \{0\})$, 并且最优解对应于上式中的最优顾客

子集 U^* 。然而,由于这种方法需要考虑相当庞大的状态数,故只能精确求解规模非常小的问题。Christofides 等通过运用状态空间松弛技术大大减少了状态数,使动态规划算法的性能得到了很大程度上的改善^[31]。他们的实验表明,该方法可有效求解顶点数不超过 50 的有容量约束 VRP。

货物流模型和车辆流模型是整数线性规划模型中的另外两种:货物流模型明确地考虑通过每条路径的货物量,车辆流模型则体现了系统中车辆的最优运行方案。由于 VRP 中的许多问题既可采用车辆流模型,又可采用货物流模型,因此这里将这两种模型合并在一起统称为流模型。

在以往的研究中,采用的流变量主要有以下几类:

$$(1) \quad x_{ij} = \begin{cases} 1, & \text{若某辆车在弧}(i, j)\text{上行驶} \\ 0, & \text{否则} \end{cases}$$

$$(2) \quad x_{ijk} = \begin{cases} 1, & \text{若车辆}k\text{在弧}(i, j)\text{上行驶} \\ 0, & \text{否则} \end{cases}$$

$$(3) \quad w_{ij} = \text{通过弧}(i, j)\text{的货物量}$$

$$(4) \quad w_{ijl} = \text{通过弧}(i, j)\text{驶往顾客}l\text{的货物量}$$

$$(5) \quad z_{ik} = \begin{cases} 1, & \text{若顾客由车辆}k\text{服务} \\ 0, & \text{否则} \end{cases}$$

使用双下标和三下标变量各有优劣。双下标变量的特点是简便,涉及的变量较少,然而却不能反映不同车辆的成本与特征,因此只适用于车辆类型相同的情况。Desrosiers 等^[32]、Laporte 等^[33]分别采用了双下标变量来建模,其中 Laporte 等提出了一种约束松弛算法,应用于约束条件较松的问题,效果较好。

相比之下,三下标的 VRP 模型更通用一些,然而涉及的变量一般较多。Martello 等研究了有容量和时间窗约束 VRP 的三下标模型,并且基于 Bender 分解,通过交替求解一般指派问题和有时间窗的旅行商问题来求解该问题^[34]。

尽管人们对确定性 VRP 的精确算法作了相当多的研究,但由于所有的 VRP 都属于 NP 难题,精确计算所花费的成本随问题规模的增大呈指数增长,到目前为止,还未有一种精确算法能够获得顶点数超过 100 的问题的解。因此在生产和生活实践中,为了能以合理的计算成本获得问题的近优解或满意解(从实用角度出发,事实上已能满足要求),人们不得不将注意力集中在设计启发式算法上。

众多专家学者通过对经验的归纳推理和实验分析的结论,结合自己的感知和洞察,提出了大量的启发式方法。由于这些方法精彩纷呈,既体现了自然科学的逻辑性,又体现了人类智慧的能动性,因此被认为是“科学与艺术的完美结合”。迄今为止,提出的启发式算法可大致分为以下几类:

(1)先分组后安排线路的方法。这种方法是先按顾客需求进行分组,然后设计每一组的车辆路径。

(2)先安排线路后分组的方法。这种方法首先构造包含所有顾客的长路径,然后按照某种规则将长路径分成数条短而可行的路径。

(3)节约-插入法。这种方法首先计算出在当前路径的各个可行位置插入不在当前路径中的顾客使目标函数的增加值(或节约值),然后根据目标函数增加值(或节约值)的升序(或降序)排列,逐个插入各顾客,如 Clarke-Wright 节约算法^[35],Solomon 插入法^[36]等。

(4)改进-交换法。在保持路径可行的情况下,每次迭代产生一个解用以替代原来的解,力图使目标函数值得以改进,一直继续到不能再改进目标函数时为止,如 Lin 提出的 2-opt 和 3-opt 交换^[37]等。

(5)基于数学规划的方法。把问题表达成一个数学规划模型,根据其模型的特殊构形,利用一定技巧进行分划,进而求解易于处理的子问题。

(6)亚启发式算法。进入二十世纪九十年代,亚启发式算法的应用成为 VRP 研究中的新趋势。神经网络(Neural Network)、模拟退火算法(Simulated

Annealing)、禁忌搜索算法(Tabu Search)和遗传算法(Genetic Algorithm)等启发式算法已经较成功地解决了确定性 VRP 中的一些问题。

以上对启发式算法的分类并不是绝对的,某一特定算法可能涉及以上几类方法。对于具体问题,往往要考虑到问题的具体特征,才有可能地构造出高效省时且行之有效的启发式算法。

国外从实用化角度开展的研究也比较多,研究者将模型和算法方面取得的进展应用于很多具体问题,如:燃油补充问题,食品和饮料配送问题,工业垃圾收集问题等。由此诞生了一些为企业车辆路径提供服务的专业化公司,开发了各具特色的车辆路径软件,比较著名的有:美国 ESRI 公司的 ArcInfo 系统, Roadnet 科技公司的 Roadnet5000 系统, Routesmart 科技公司的 Routesmart 系统, Optrak 软件公司的 Optrak 系统。

二、随机性 VRP 的研究现状

在日常生活和工作中,人们经常会遇到一些不能确切地预计其后果,但是能够知道其后果出现规律的随机事件,在 VRP 中也经常会出现这种事件。在构造车辆路径之前,与问题有关的某些信息并不完全知晓,仅能根据历史资料或市场调查获得某些信息的统计规律,也就是 VRP 中的某些要素是随机的。由于这类问题有着广泛的应用背景,并且其求解方法和解的特征与确定性 VRP 有很大区别,所以近些年来引起人们的广泛关注。

尽管目前对随机 VRP 的研究还只限于随机的顾客、随机的需求量和随机的旅行时间等,但在实际中出现的随机性因素还远远不止这几种,引起随机 VRP 的因素可以概括为下面五类:

(1)需求的随机性,如需求量、需求时间的随机性,目前的研究主要集中在由该类因素导致的随机性 VRP;

(2)提供服务的车辆、司机及其它设施的随机性;

(3)网络性能的随机性,因天气变化、交通堵塞等原因可能使旅行时间或网络容量产生随机性变化;

(4)未来时间段网络管理和运作的不确定性。这类问题是相当复杂的,因为预测未来某一时刻系统管理与运作发生的变化是十分困难的;

(5)提供的数据具有偏差。必须承认,所有模型的输入数据都可能存在误差,如何处理这些误差仍是一个难题,与(4)相同,目前还没有这方面的研究。

随机需求量的车辆路径问题(VRPST)是指尽管确切知道需要服务的顾客,但却不能获得其准确的需求量,仅能获得需求量的概率分布。在 VRPST 研究的一些重要文献中,Stewart 等提出一种机会约束模型和两种补偿规划模型^[38]。此外,该文还考虑了多种需求分布,并对两种启发式算法进行测试,一种基于 C-W 节约算法,一种基于 Lagrangean 松弛算法。

随机顾客和需求量的车辆路径问题(VRPSCD)将 VRPST 和 VRPSC 结合在一起。VRPSCD 是一个非常困难的问题。甚至计算目标函数的值也是相当困难的。Gendreau 等提出求解该问题的一种禁忌搜索算法,这是禁忌搜索在随机 VRP 领域的第一次应用。对于顶点数从 6-46 的 825 个测试问题,该算法可以得到其中 89.45%问题的最优解^[39]。

由于交通事故、天气变化等因素常会引起交通密度的变更和车辆行驶速度的变化,进而导致旅行时间的改变。在这种情况下,各顶点间的旅行时间可考虑为具有某种统计规律的随机变量,这样的路径问题称为随机旅行时间的车辆路径问题(VRPST)。目前对 VRPST 的研究很少。

在随机 VRP 中, 还存在着—类更为复杂的问题, 在这些问题中, 信息随着时间的变化而发生更新, 具有“动态”的特点, 故称它们为动态随机 VRP。该问题的主要特征如下^[40]:

(1) 制定路径和执行路径计划的过程中可接受新信息。在静态 VRP 中, 所有信息都是已知的, 故在路径制定和执行时不考虑出现新的顾客、道路堵塞、天气变化等情况, 因此, 传统的静态 VRP 是建立在很强的假设条件基础上的, 具有很大的局限性。而动态 VRP 不仅考虑已知信息, 还考虑可能在路径制定和执行的任一时刻出现的新信息, 具有很强的灵活性。

(2) 未来信息可能是不精确或未知的。在静态情况中, 几乎没有“过去”“现在”和“未来”之分。而在动态问题中, 在实时信息出现之前, 并不能确定地掌握这些信息, 可能只掌握它们的统计规律或模糊规律, 甚至在有些情况下, 连这样的信息也不可能得到。

(3) 要求快速响应接受的新信息。在静态路径系统中, 对处理信息的时间要求并不很高, 人们通常可以花费几个小时获得要求解问题的结果。而在动态路径系统中, 需要实时处理新接收的信息, 通过启发式方法快速计算, 重新安排车辆路径, 并将结果及时通知司机和顾客。

(4) 目标函数往往非常复杂。在静态 VRP 中, 要求优化的目标函数比较简单, 容易确定, 一般是运输距离最短、时间最少或费用最小等。而动态 VRP 的计划期可能是无限的, 不能使用静态 VRP 基于单阶段的目标, 只能选择诸如通过率、服务率等作为目标。

由以上的特点, 不难明了动态 VRP 具有很大的不确定性, 动态 VRP 的研究必然和非确定性 VRP, 尤其是随机 VRP 的研究交织在一起。在以往的研究中, 动态随机 VRP 的建模方法主要有 Markov 决策、排队论和网络方法。

由于动态随机 VRP 的问题规模一般较大, 而且算法需要使实时信息在较短时间内得到响应, 所以一般的最优化方法往往不能适应, 人们的研究主要集中在启发式算法上。从求解策略来看, 动态随机 VRP 的算法可分为两大类: 一类是重新优化策略, 另一类是局域优化策略。

重新优化策略实际上就是静态问题的动态求解, 即一旦接收到实时信息, 就从头开始重新寻找最优车辆路径。

VRP 本身就是 NP 难题, 计算量非常大, 如果每接收一项实时信息就重新进行优化, 那么所需的计算量简直不可想象; 况且, 在许多情况下必须对信息进行快速处理, 可能没有重新优化所需的时间与其它资源。因此, 在这方面的研究较少。

与重新优化策略不同, 局域优化策略是事先根据已知的信息制定初始路径, 当接收到实时信息后, 用局域方法改进初始路径。尽管局域优化策略获得的路径计划可能劣于重新优化策略, 但是却能节约大量的计算, 因而更适用于实际的车辆调度系统。由于这种原因, 对运用局域优化策略的算法研究较多, 下面的算法都采用了这种策略^[40]。

Wilson 于 1971 年提出了求解 dial-a-ride 的插入法, 插入的原则是在所有可行插入位置上最小化从需求发生到计划装载的延迟时间、从装载点到卸载点之间的行驶时间, 以及计划装载和承诺装载间的时间差。在 Wilson 与其合作者在后来的研究中, 又对该插入法作了改进, 一旦接收到新需求, 并不马上进行指派, 而是等待未来的需求到来, 以降低插入法的短视行为。

类似地, Ray 等人提出一种插入法应用于运送残疾人问题。在这种情况下,

相当多的需求事先已知, 初始路径根据这些需求构造。算法先根据时空近似度构造顾客的子簇; 在每个子簇中, 用插入法构造路径; 最后, 通过使顾客从一条路径转移到另一条路径, 在尽可能减小路长的原则下, 使解得到改进。

Madsen 等人也研究了运送老年人和残疾人的问题, 提出了一种插入方法。算法先用 Jaw 等人提出的修正插入法静态安排车辆路径, 然后根据新顾客和已插入顾客之间可能存在的利益冲突抽象出一个冲突函数, 并以最小化冲突函数作为新顾客的插入准则, 按序列模式处理实时需求。

Gendreau 等构造了一种带有适应性存储单元的禁忌搜索算法, 用于并行求解信使服务问题。算法按主-从模式构造, 主进程控制适应性存储单元(用于存储在搜索过程中所发现的最好几条路径); 从进程执行以 CROSS 交换作为邻域结构的禁忌搜索算法。当接受一项新的服务需求时, 从进程停止, 将目前从进程所能搜索到的最好解送至主进程的适应性存储单元, 利用传统的插入方法, 将新需求插入适应性存储单元内的每个解中。在适应性存储单元得到更新后, 将其存储的解传递给从进程, 重新启动禁忌搜索算法进行优化。

2.3 TSP 问题的国内外研究综述

TSP (Traveling Salesman Problem) 问题实际上是 VRP 问题的一个特殊情况, 它被称为旅行商问题, 也有称为旅行推销员问题或邮递员问题。

2.3.1 问题描述

住在某城市的推销员, 要访问 $N-1$ 个城市, 希望每个城市都访问一次且只访问一次, 然后回到出发点。他应该按怎样的次序访问这些城市, 以使走过的总行程或时间最短, 或是旅费最省? 这样的问题叫旅行商问题或旅行推销员问题, 亦称货郎担问题, 有些地方还称为推销员问题。对于现实问题, 由于限制条件的增加, TSP 可衍生出许许多多相关的问题。

TSP 问题在图论的意义下就是所谓的最小 Hamilton 圈问题, 以图论的形式描述为: 在图 $G=(V, E)$, V 是点(城市)的集合, E 是边的集合, $E=\{(i, j) | i, j \in V\}$ 。

点 i 与 j 的欧氏距离为 d_{ij} , 设 $d_{ij}=d_{ji}$ 。目标是找到一个长度最小的闭合回路,

使得访问每个点一次且仅一次, 这条闭合回路也称作哈密顿回路。

此外, TSP 从组合优化的角度描述, TSP 是一个有限集合下的最优化问题:

有 n 个城市 A_1, A_2, \dots, A_n 。已知从 A_i 城到 A_j 城的旅费(或距离)为 c_{ij} , 其 n

阶矩阵为 $C=(c_{ij})$ 。推销员从某城出发, 遍历其它 $n-1$ 个城市各一次, 最后返回出发点, 求总旅费最少(或总距离最短)的路径。

则它的数学模型表示如下:

目标函数:

$$\text{Min} \sum_{i \neq j} c_{ij} x_{ij} \quad (2.2)$$

S.T:

$$\sum_{j=1}^n x_{ij} = 1, i = 1, 2, \dots, n, \quad (2.3)$$

$$\sum_{i=1}^n x_{ij} = 1, j = 1, 2, \dots, n, \quad (2.4)$$

$$\sum_{i, j \in s} x_{ij} \leq |s| - 1, 2 \leq |s| \leq n - 2, s \subset \{1, 2, \dots, n\} \quad (2.5)$$

$$x_{ij} \in \{0, 1\}, i, j = 1, \dots, n, i \neq j$$

其中, c_{ij} 表示第 i 城市到第 j 城市的距离, 决策变量 $x_{ij} = 1$ 表示推销员行走的路线包含从城市 i 到城市 j 的路径, $x_{ij} = 0$ 表示推销员没有选择这条路。 $i \neq j$ 的约束可以减少变量的个数, 使得共有 $n \cdot (n-1)$ 个决策变量。约束 2.4 要求推销员从城市 i 出来只有一次, 约束 2.5 要求推销员走入城市 j 只有一次。约束 2.4 和约束 2.5 表示每个城市经过一次。仅有 2.4 和 2.5 的约束无法避免回路的产生, 一条回路是由 k 个城市和 k 条弧组成, 因此, 约束 2.6 使得推销员在任何一个城市子集中不形成回路, 其中 $|s|$ 表示集合 s 中元素个数^[41]。

因为集合 s 是一个有限集合, 因此我们可以用枚举的方法证明最优解 x^* 一定存在, 但是我们知道随着问题规模的扩大, 用枚举法显然不现实。

- 当 $c_{ij} = c_{ji}$ ($i, j = 1, 2, \dots, n$) 时, 上述问题称为对称型 TSP; 当对所有

$i, j, k = 1, 2, \dots, n$, 有不等式 $c_{ij} + c_{jk} \geq c_{ik}$ 成立时, 问题被称为满足三角形不等式的 TSP, 简写成 ΔTSP 。

2.3.2 TSP 的历史

关于 TSP 问题的提出可以追溯到 17 世纪, 爱尔兰数学家 Sir William Rowan Hamilton 和英国数学家 Thomas Penyngton Kirkman, 这两个人早期对于 TSP 的工作可以从“Graph Theory 1736-1936”(N. L. Biggs, E. K. Lloyd, and R. J. Wilson, Clarendon Press, Oxford, 1976.)看到。

TSP 问题的一般描述是由哈福大学的 Karl Menger 在 20 世纪 30 年代提出的。后来这个问题被普林斯顿大学的 Hassler Whitney 和 Merrill Flood 进一步精确的定义。

TSP 已经被证明是 NP 完全问题, 是典型的易于描述, 但是最优化求解很难的问题。主要原因是所谓的“组合爆炸”现象。例如, 对于置换排列描述的 n 城市 TSP 问题有 $(n-1)!/2$ 种不同的排列, 状态数量随着问题规模呈超指数增长。 $n=20$ 的时候, 使用穷举法求解, 即使计算机每秒能够处理 1 亿种排列, 也需要大约几百年的时间。^[41]

2.3.3 TSP 问题的分类

按问题对应到图的类型, TSP 可以分为两类:

1. 城市间的距离都是对称的, 它对应的是图论中的无向图;
2. 两个城市间的距离是非对称的, 它所对应的是图论中的有向图。

按问题本身的限制条件的强弱, 主要有三类:

1. 不做任何限制(但是一般都要求城市间的费用不为负数), 只是给出距离矩阵, 求最小回路;
2. 要求距离间要满足三角不等式;
3. 定义在欧式平面上的 TSP, 即 Euclid TSP, 它给出每个点在欧式平面上的坐标, 而城市间的距离就是以他们的欧式距离来定义。

按问题的多项式可解性上分, TSP 可以分为两类:

1. 目前已经知道有多项式时间算法可解的, 比如其距离矩阵满足特定的条件(Demidenko 条件、Kalmanson 条件、Supnick 条件)等;
2. 目前尚没有发现多项式时间算法可解的, 而研究热点是如何寻找更多的多项式时间可解的情形。

除了经典 TSP 外还有很多扩展形式, 常见的有:

- 最小 Hamilton 路问题。起点和终点不同的 TSP。
 - 瓶颈问题 (bottleneck TSP。即求最大化问题, 目标函数为 $\max_{i \neq j} c_{ij} x_{ij}$ 的 TSP。
 - 有向 TSP (asymmetric TSP)。距离矩阵非对称的 TSP。
 - 多人 TSP (multi-person TSP)。由多人完成环游的 TSP。
- 多目标 TSP (multi-objective TSP)。既求多个目标值 (考虑多方面因素最佳折中解) 的 TSP。

2.3.4 关于 TSP 问题的国内外研究现状

TSP 问题实际上是 VRP 问题中一个具有代表性的特殊问题, TSP 问题是组合优化领域中的一个典型问题, 涉及求多个变量的函数的最小值。虽然它陈述起来很简单, 但求解却很困难, 它一直是运筹学中最富挑战性的问题之一, 并且已经被证明是 NP 完全问题。对于具有 n 个城市的 TSP 问题, 其可能的路径数目为 $(n-1)!/2$, 至今尚未找到有效的求解方法, 在理论上枚举法可以解这一问题, 但是当 n 较大时, 解题的时间消耗会使枚举法显得没有任何实际价值。因此寻求一种求解时间短, 能满足实际问题精度要求的解, 成为解决该问题的主要途径。

TSP 问题的求解, 一直以来倍受人们的关注。对于这样一个典型的、易于描述却难以处理的 NP 难题, 有效地解决它在可计算理论上有着重要的理论价值。并且, TSP 问题是诸多领域内出现的多种复杂问题的集中概括和简化形式。该问题在工程领域有着广泛的应用, 如网络通讯、电气布线、管道铺设、货物运输、加工调度等, 另一方面, 很多实际应用问题, 如印制电路板的钻孔路线方案、连锁店的货物配送路线等, 经过简化处理后, 均可建模为旅行推销商问题, 因此, 快速、有效地解决 TSP 问题有着极高的实际应用价值, 也就吸引了众多领域的学者对它进行研究。研究方法种类繁多, 研究成果不断涌现。尽管 TSP 仍未找

到最优解，但是求解它的算法逐渐在改进。正是由于 TSP 问题的代表性求解难度，激发了人们对优化技术的研究和对 TSP 问题本身的挑战，TSP 已经成为优化技术成功的主要体现之一。几十年以来，随着对优化技术的深入研究，以及计算机处理速度和内存容量的快速增长，人们取得了一个又一个的纪录。

1980 年 Crowder 和 Padberg 求解了 318 个城市的问题。

1987 年 Padberg 和 Rinaldi 将这个城市数增加到了 2392 个。

1992 年美国 Rice 大学的 CRPC 研究小组用 50 台工作站使用了基于“cutting planes”算法解决了 3038 个城市的问题，被《发现杂志》评为当年的前 50 条科学新闻。

1994 年, Applegate, Bixby, Chvatal 等人使用若干台 SPARC 工作站组成的机群用了 3-4 年的 CPU 时间解决了 7397 个城市的 TSP 问题。

1998 年, CRPC 研究小组使用三台 Digital AlphaServer 4100s (12 个处理器)组成的集群和 32 台 Pentium-II 个人计算机解决了美国 13,509 个城市组成的 TSP 问题。

1998 年, 美国加利福尼亚大学 Dan Gusfield^[42]根据出度、入度均为 2 的有向图 G 中有一条 Euler 路当且仅当有一条 Hamilton 路这一特性, 提出用 Greedy 算法求解出度、入度均为 2 的 TSP。

2000 年, 周培德^[43]用点集凸壳的多项式时间算法解决了 31 个顶点的中国货郎担问题, 其时间复杂度为 $O(n^2 \lg n)$ 。

2000 年, 澳大利亚 Vladimir G. Deineko^[43]证明了距离满足 Demidenko 矩阵时, MaxTSP (指遍历所有城市的最大权 Hamilton 圈) 是 NP-难题, 而之前已证 TSP 在此条件下是多项式时间可解的。

2003 年 2 月, Hisao Tamaki 使用了路径融合 Lin-Kernighan 启发 (LKH) 的变种相结合的方法发现了 TSPLIB 中 pla33810 的一个次优解。

2004 年 2 月, Keld Helsgaun 发现了 pla85900 问题的一个次优解。此外, 他又于 2003 年 12 月发现了 7,516,353,779 个节点的世界 TSP 问题的一条比较好的解。这是目前为止已知的求解规模最大的 TSP 问题。

不仅 TSP 本身成为人们研究的热点, 而且由于 TSP 问题的代表性, 许多新的算法, 理论和思想在被提出后也常常使用 TSP 作为测试其自身性能的标准。因此, TSP 成为多种启发式的搜索、优化算法的间接比较标准。

20 世纪 80 年代以来, 随着模拟退火, 进化计算, 神经网络的兴起, 人们逐步关注使用这些现代优化算法求解 TSP 问题, 并且取得了意想不到的成果。随着近年来群集智能 (Swarm Intelligence) 的提出, 美国的圣塔菲研究所的一些科学家提出了使用蚂蚁算法求解 TSP 问题, 获得了可喜的成果。

2.4 求解 TSP 和 VRP 问题的算法综述

作为组合优化的一个经典问题, TSP 问题和 VRP 问题吸引了广大学者对它进行研究。这一章里我们总结了相关的一些计算方法, 主要分为两大类: 一类是精确算法 (完全算法); 另一类是近似算法或启发式算法 (不完全算法)。

完全算法能保证完全搜索问题的整个解空间, 从而找到最优巡回, 但需要消耗 $O(n!)$ 级的运算时间; 有些完全算法虽然运用一些精巧的技术来减少搜索空间, 但本质上还是进行全局搜索, 并没有降低运算时间复杂度。

不完全算法不能保证搜索问题的全部解空间, 所以也不能保证能够找到最优解, 甚至在某些实例上连解都得不到, 但它们与完全算法相比却具有运算时间上的优势, 即它们的运算时间复杂度都只是多项式, 并不随着输入规模的扩大产生“组合爆炸”。这类算法采用的是启发式策略来指导搜索, 普遍比完全算法要快。不完全算法一般又可以细分为近似算法和改进算法。

下面将分别对它们分别做简要介绍^[44]。

2.4.1 精确算法

1. 穷举搜索法

这是求解 TSP 问题的最早的一种算法, 是一种最简单但是最费时的方法, 因为它要把所有的路径方案的长度都列出来, 从中选择一个最短的。穷举法的优点是找到的优化路径一定是全局最优的解, 但其致命的缺点为惊人的计算开销。用该法计算 n 个城市的 TSP 问题, 所要计算的路径为 $(n-1)! / 2$ 条。

2. 贪婪法

在组合算法中, 将每一步都取局部最优的求解方法成为贪婪法。由于贪婪法的局限性, 它只能求解“单峰”的情况, 对于“多峰”的复杂情形则不能得到全局最优解。因此, 贪婪法用在 TSP 问题上一般可以得到局部最优解。

3. 线性规划算法

主要采用整数规划中的割平面法, 即先求解模型中由前两个约束构成的松弛 LP 问题, 再通过增加不等式约束产生割平面, 逐渐收敛到最优解。

4. 动态规划算法

动态规划法的主要特点, 是将一个问题分为若干互相联系的阶段, 每个阶段进行决策优化。其理论依据为 Bellman 最优性原理。

记 S 为集合 $\{2, 3, \dots, n\}$ 的子集, $k \in S, C(S, k)$ 为从 1 出发遍历 S 中的点并终止在 k 的最优行程。当 $|S|=1$ 时, $C(\{k\}, k) = d_{1k}, (k = 2, 3, \dots, n)$, 当 $|S| > 1$ 时根据最优性原理, 可将 TSP 的动态规划方程写成如下形式:

$$C(S, k) = \min_{j \in S - \{k\}} [C(S - \{k\}, j) + d_{jk}] \quad (2.6)$$

按方程规则可逐步迭带求解。

由于动态规划算法的时间复杂度为 $O(n^2 \cdot 2^{1^n})$, 空间复杂度为 $O(n \cdot 2^n)$, 因此一般除了很小规模的问题外, 几乎不采用此方法。

5. 分支定界法

分支定界法是一种应用范围很广的搜索算法, 它通过有效的约束界限来控制搜索进程, 使之能向着状态空间树上有最优解的分支推进, 以便尽快找出一个最优解。该方法的关键在于约束界限的选取。不同的约束界限, 可形成不同的分支定界法。

(1) 以分派问题为界

通过求解相应的分派问题, 得到 TSP 问题的一个下界, 以此进行分支定界搜索。这是一种使用较多的分支定界法。

(2) 以匹配问题为界

通过求解相应的匹配问题, 得到 TSP 问题的一个下界, 以此进行分支定界搜索。该方法适用于对称型 TSP。

(3) 以最小 1 树为界

通过求解相应的最小 1 树问题, 得到 TSP 的一个下界, 以此进行分支定界搜索。

虽然分支定界法对于较大规模的问题并不十分有效, 可有时却被用来求解近似解。而且, 将分支定界法与一些启发式算法相结合, 还常常能获得一些意外的成功。

2.4.2 近似算法

由于精确算法所能求解的问题规模非常有限, 实际中使用的往往是多项式阶数的近似算法或启发式算法。但它们基本上仅适用于对称型 ΔTSP 。此类算法的好坏用 $C/C^* \leq e$ 来衡量, C 为近似算法所得到的总行程, C^* 为最优总行程, e 为最坏情况下, 近似解与最优解的总行程之比的上界值。

通常有下列一些近似解法:

1. 插入算法

插入算法提出了一个不同的构造巡回的启发式算法。它首先开始于连接两个结点的巡回, 然后把剩余的结点逐个地加入到巡回中, 在加入结点的过程中要使得巡回费用的增加最小。从哪两个结点开始, 更重要的是在每一个步骤选择哪个结点插入, 都会影响最后的结果。

按插入规则的不同分为若干类:

- (1) 最近插入法。最坏情况: $e=2$; 时间复杂度: $O(n^2)$ 。
- (2) 最小插入法。最坏情况: $e=2$; 时间复杂度: $O(n^2 \lg n)$ 。
- (3) 任意插入法。最坏情况: $e = 2 \lg n + 0.16$; 时间复杂度: $O(n^2)$ 。
- (4) 最远插入法。最坏情况: $e = 2 \lg n + 0.16$; 时间复杂度: $O(n^2)$ 。
- (5) 凸核插入法。最坏情况: e 未知; 时间复杂度: $O(n^2 \lg n)$ 。
- (6) 最优插入法。最坏情况: $e=2$; 时间复杂度: $O(n^3)$ 。

在实际中, 通常最好的插入算法是最远插入法。对该算法, 我们连接一些高费用边的两个端点形成最初的巡回。对每一个未插入的结点 V , 我们计算出该结点插入到当前巡回中巡回费用的最小费用, 然后选择那个费用最大的结点作为下一个结点继续插入。这初看起来和直觉不一样, 然而在实际中的效果往往令人满意。这可能是因为最终巡回的粗糙原型要较早得到, 在以后的步骤, 只需要有相对较小的改动。

2. 最近邻算法

该算法的基本思想是: 从任意一个结点开始, 访问当前未访问过的结点中最

近的结点，当所有的结点都访问过时回到最初的开始结点。

具体实施中，可以把出发点取遍 V 中各点而得到多个解，从中选择最好的一个，但此时的时间复杂度增加了 n 倍。先任取一出发点，再依次取最近的点加入当前解中直至形成回路解。

最坏情况： $e = (\lg n + 1)/2$ ；时间复杂度： $O(n^2)$ 。

3. Clark & Wright 算法

最坏情况： $e = 2\lg n/7 + 5/21$ ；时间复杂度： $O(n^2)$ 。

4. 双生成树算法

最坏情况： $e = 2$ ；时间复杂度： $O(n^2)$ 。

5. Christofides 算法

最坏情况： $e = 3/2$ ；时间复杂度： $O(n^3)$ 。

6. r -opt 算法

一种局部改进搜索算法。对给定的初始回路，通过每次交换 r 条边来改进当前解。

最坏情况： $e = 2(n \geq 8, r \leq n/4)$ ；时间复杂度： $O(n^r)$ 。

7. 混合算法

用某个近似算法求得初始解，然后借助一个或若干个 r -opt 算法对解加以改进。这种混合型的算法往往能获得较好的解，但很耗时，一般仅在对解有较高要求时采用。

8. 概率算法

最坏情况： $e = 1 + (\text{任意给定正数})$ ；时间复杂度： $O(n \lg n)$ 。

2.4.3 改进算法

从一初始解开始，通过对当前的解进行反复地局部扰乱以达到较好的解。

1. 神经网络算法

八十年代中后期，美国、日本等国家出现了一股神经网络热潮，许多从事脑科学、心理学、计算机科学以及电子学等方面的专家都在积极合作，开展这一领域的研究。其早期思想源于四十年代，由于受到 Von Neumann 串行处理体系的限制，一直进展不大，直到 1982 年，美国生物物理学家 Hopfield 提出了一种离散神经网络 (ANN) 模型，才被认为是一个重大突破。而 Hopfield 于 1984 年又提出一种连续时间神经网络 (HNN) 模型，并由容易实现的电子线路所构成。对于 HNN，给出适当的初始条件，在状态空间里反复使其更新状态，网络的能量随时间推移单调地减小，状态向着平衡状态的方向更新。最后，网络的能量减至全局最小或局部最小，其状态稳定在某个平衡状态。利用 HNN 模型在状态空间里的这种能量最小（极小）化特性，可将它应用于 TSP 问题的求解。

该方法的基本思想是通过对神经网络引入适当的能量函数，使之与 TSP 的目标函数相一致来确定神经元之间的联结权，随着网络状态的变化，其能量不断

减少，最后达到平衡时，即收敛到一个局部最优解。

要解 n 城市的 TSP 问题，要把问题映射到一个神经网络上。可以使用 $n \times n$ 神经元矩阵。矩阵中的每个元的状态只能为 0 或 1，神经元的状态用 V_{si} 表示，

$V_{si}=1$ 表示城市 x 在路径中第 i 个位置出现。一次有效路径使每行每列有且仅有一个元素为 1，其余为 0。为了最终解决 TSP 问题，必须构成这样的神经网络：在网络运行时，计算能量降低，网络稳定后其输出状态表示城市被访问的次序。网络能量的极小点，对应于最佳（或较佳）路径的形成。其解决问题最关键的一步，是构造能量函数。金海和等曾提出了基于 HNN 的多城市旅行商问题的求解算法^[45]。它是把 HNN 学习算法作为基本算子，对城市群体按一定的规则进行有效的分割、计算和连接，来寻找巡回路径的最优解或满意解，取得了一定的成果。但人工神经网络算法的能量函数中的多个参数选择往往要求应用人员有很高的技术性和对问题有较深的了解，而且它的能量函数实际上是罚函数，这势必会造成计算量的增加。

2. 遗传算法

遗传算法（Genetic Algorithms——GA）是模拟生物在自然环境中的遗传和进化过程而形成的一种自适应全局优化算法，是美国 Michigan 大学的 Holland 教授于 1975 年首次提出，起源于 60 年代对自然和人工自适应系统的研究。70 年代 DeJong 基于遗传算法的思想在计算机上进行了大量的纯数值函数优化计算实验。在一系列研究工作的基础上，80 年代由 Goldberg 进行归纳总结，形成了遗传算法的基本框架。Goldberg 和 Michalewicz 进行了大量的研究工作，并成功地将它应用到各种领域的优化问题。

遗传算法的早期应用主要围绕组合优化问题求解，如煤气管道的最优控制，TSP 问题等，近年来迅速扩展到机器学习、设计规划、神经网络优化、核反应堆控制、喷气发动机设计、通信网络设计、人工生命等领域，显示了遗传算法应用的巨大潜力。

近年来，遗传算法从理论到实际都已经取得了许多重要成果。由于它具有良好的全局搜索能力，是目前解决各种优化问题的最有效的方法，已经成为研究热点。遗传算法就其本质来说，主要是处理复杂问题的一种鲁棒性强的启发式随机搜索算法。因此遗传算法在 TSP 问题求解方面的应用研究，对于构造合适的遗传算法框架、建立有效的遗传操作以及有效地解决 TSP 问题等有着多方面的重要意义。但是遗传算法存在的一些问题也是不同忽视的，由于遗传算法对自己的系统参数要求较高，必须选择种群数目、杂交概率、变异概率等，选择的参数好，就会得到较好的实验结果；反之，就会使算法的收敛速度变慢，而且经常收敛于局部极小点，甚至可能导致实验的失败。

3. 自适应蚁群算法^[46]

人工蚁群算法是人们受到对自然界中真实的蚁群集体行为的研究成果的启发而提出的一种基于种群的模拟进化算法，属于随机搜索算法，由意大利学者 M.Dorigo 等人首先提出。该算法充分利用了蚁群搜索食物的过程与 TSP 问题之间的相似性，通过人工模拟蚂蚁搜索事物的过程（即通过个体之间的信息交流与相互协作最终找到从蚁穴到食物源的最短路径）来求解 TSP。

象蚂蚁这类群居昆虫，虽然单个蚂蚁的行为极其简单，但由这样的单个简单的个体所组成的蚁群却表现出极其复杂的行为，能够完成复杂的任务。蚁群之所

以表现出复杂有序的行为，个体之间的信息交流与相互协作起着重要的作用。蚂蚁在运动过程中，能够在它所经过的路径上留下该种物质，而且蚂蚁在运动过程中能够感知这种物质的存在及其强度，并以此指导自己的运动方向。因此，由大量蚂蚁组成的蚁群的集体行为便表现出一种信息正反馈现象：某一路径上走过的蚂蚁越多，则后来者选择该路径的概率就越大。

用蚁群算法求解 TSP 时，我们不难发现：蚁群算法的主要依据是信息正反馈原理和某种启发式算法的有机结合，这种算法在构造解的过程中，利用随机选择策略，这种选择策略使得进化速度较慢，正反馈原理旨在强化性能较好的解，却容易出现停滞现象，而且它容易陷入局部最优解。

4. 模拟退火算法

模拟退火算法是近年来特别引人注目的一种适应于解大型组合优化问题的技术，算法的核心在于模仿热力学中液体的冻结与结晶或金属溶液的冷却与退火过程。在高温状态下，液体的分子彼此之间可以自由移动。如果液体徐徐冷却，它的分子就会丧失由于温度而引起的流动性。这时原子就会自己排列起来而形成一种纯晶体，这个纯晶体状态就是该系统的最小能量状态。算法的基本思想是从一定解开始的，从邻域中随机产生另一个解，接受准则允许目标函数在有限范围内变坏。它由一控制参数 t 决定，其作用类似于物理过程中的温度 T ，对于 t 的每一取值，算法持续进行“产生新解—判断—接受或舍弃”的迭代过程，对应着固体在某一恒定温度下趋于热平衡的过程。经过大量的解变换后，可以求得给定控制参数 t 值时优化问题的相对最优解。然后减小控制参数 t 的值，重复执行上述迭代过程。当控制参数逐渐减小并趋于零时，系统亦越来越趋于平衡状态，最后系统状态对应于优化问题的整体最优解。该过程也称为冷却过程。由于固体退火必须缓慢降温，才能使固体在每一温度下都达到热平衡，最终趋于平衡状态，因此，控制参数的值必须缓慢衰减，才能确保模拟退火算法最终趋于优化问题的整体最优解。模拟退火算法要从邻域中随机产生另一个解，对于 TSP 问题，它的邻域是指两条路径除局部有差别外，大多数路径相同。

解决 TSP 问题的模拟退火算法的框架为：

给定起、止“温度” T ， T_0 和退火速度 α ，初始一条路径 C_0 ；

While ($T > T_0$) do

在 C_0 的邻域内产生另一条路径 C_1 ；

计算两条路径所引起的目标函数（能量）值的变化 ΔE ；

若 $\Delta E \leq 0$ ，接受新值，若 $\exp(-\Delta E / T) > \text{rand}(0,1)$ ， $\text{rand}(0,1)$ 表示 0 到 1 之间的随机数，也接受新值，否则就拒绝；

确定新的参数值，若扰动被接受，则 C_1 替换 C_0 ，否则 C_0 不变；

若接受新值，降温 αT ，否则不降温；

END

模拟退火算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，并可以人为控制迭代次数，反复求解，因此具有很强的实用性。

2.4.4 优化算法比较

本文所研究的搬运机路径优化问题也可以看作 TSP 在车辆路径调度问题中的应用,结合了最小 Hamilton 路问题和多目标 TSP 问题,即搬运机所服务的起点和终点不是同一点的问题,而且本文的目的是综合最小化搬运机行驶总距离和最大化顾客服务水平两个目标。而且本文所研究的搬运机路径优化问题又是一个动态 VRP 问题,所以对算法计算时间上的要求较高。

由于精确算法在问题规模增大时计算时间指数增长,因此不适合本问题的求解。而大部分的近似算法缺乏通用性,每一种方法讨论的情况不尽一致,适用范围也不完全相同,对于各种不同的情况需要单独建立模型和求解,而且当节点多、约束条件复杂时,计算速度十分缓慢。在改进优化算法中,人工神经网络算法的能量函数中的多个参数选择往往要求应用人员有很高的技术性和对问题有较深的了解,而且它的能量函数实际上是罚函数,这势必会造成计算量的增加;蚁群算法本身很复杂,一般需要较长的搜索时间,而且容易出现停滞现象,即搜索进行到一定程度后,所有个体所发现的解完全一致,不能对解空间进一步进行搜索,不利于发现更好的解。考虑到本问题的特殊性,模拟退火算法和遗传算法是最有实现可能的两种算法,根据这两种算法的内在运行机理和国内外最新的研究进展,现将这两种算法性能总结如下:

表 2.1 算法比较

算法	解的特性	初始解要求	收敛速度	终止条件
模拟退火算法	对任一优化问题,具有跳出局部最优陷井而达到问题的全局最优点的能力;该算法只需根据实际问题的优化目标和约束条件,列出代价函数表达式,而不必对问题做更深入的分析 and 建模,适用面广;该算法机制简单,可操作性好,并且擅长处理不连续变量和病态方程	初始状态; 初始温度; 最大迭代次数;对初始解的要求不高	限入局部最优次数,即温度在一定范围内能量函数达到最低值次数	迭代次数;满足精度要求;退火温度达到最低
遗传算法	擅长全局搜索,鲁棒性高,并具有获得接近最优解的能力;对搜索空间和目标函数没有任何特殊的要求(如连通性、凸性等),只利用适应值信息,而不需要导数等其它辅助信息,因而适用范围更广;从问题解的串集而不是从单个解开始搜索最优解,故有隐含的并行性;对问题参数的编码(染色体)群进行进化,而不是参数本身;是随机搜索而非确定性搜索过程,且此随机搜索具有一定的方向性,可指导搜索向着最优	减少迭代次数、随机产生的初始解既要保证不会发生早熟也不会使搜索空间太大,因此对初始解的要求较高	初始解、复制概率、交配概率、变异概率	迭代次数;满足精度要求

	解前进, 故其效率远远高于一般的随机方法;性能随问题规模的增大而降低, CPU 运算时间长;在遗传操作过程中, 有可能在串中丢失一些重要性质, 而出现“早熟”现象			
--	---	--	--	--

从表 2.1, 可以看出模拟退火算法是一种可以跳出局部极值的有效方法, 它既能抑制遗传算法的早熟现象, 又克服了其局部寻优能力较差的缺点。而且模拟退火算法的参数容易得到控制, 只要参数设置恰当, 它得到最优解的收敛速度能够得到控制, 从而有效的提高路径规划的计算速度。因此, 本人将模拟退火算法作为研究的算法。

2.5 研究中存在的问题

尽管对 VRP 和 TSP 的研究在模型和算法方面已有了许多成果, 但是由于该问题涉及的要素众多, 而且为了满足人们日益变化的需要, 还将融合其它新的要素, 仍有很多重要问题值得进一步研究, 这主要体现在:

(1) 实际中引起不确定性的因素很多且交叉影响, 而目前的研究却主要集中在由单一不确定性因素(尤其是需求的不确定性)引起的 VRP, 未综合考虑车辆、顾客、路况等各种不确定性, 与广泛的实际应用尚有距离。

(2) 在一些应用场合, 问题的规模通常很大, 而现有算法远不能满足实时动态调度车辆路径的需要, 有必要研究有效的快速算法。鉴于精确算法通常需要较长的计算时间, 如何设计和使用启发式算法值得进一步研究和发展。

(3) 现实中, 人们需求各异, 甚至相互冲突, 比如顾客满意度的提高和运作成本的降低就可能是一对矛盾, 因此在决策中需要更多地考虑多目标的情况, 但 VRP 的以往研究主要集中于单一目标问题, 对多目标问题的研究尚显不足。

(4) 现代物流更加强调系统中各个环节的协调性, 以便最大程度地提高流通效率和降低物流成本, 为满足这一需要, 有必要加强 VRP、TSP 与库存、选址等其它物流决策的综合研究。

第三章 模拟退火算法的应用分析

3.1 集装箱堆场运营管理分析

在集装箱空箱堆场管理中, 集装箱主要用来存放卸货后的空集装箱, 并进行必要的维护来保证下一次装货的需要, 因此, 集装箱堆场的作业任务主要有:

- 1) 集装箱的设备交接单(EIR, Equipment Interchange Receipt)登记: 当集装箱存放时, 在堆场信息管理系统中登记进场集装箱的资料;
- 2) 集装箱状态检验: 当集装箱存放时, 由堆场技术人员对集装箱的状态进行目视检验, 以鉴别进场集装箱处于完好状态(AVailable)或损坏状态(DaMaGe), 并判别出是使用过程中的损坏、还是箱子的自然损坏;
- 3) 集装箱作业的收费: 针对集装箱的作业要求和维护需求, 向使用单位收取约定的费用;
- 4) 集装箱的装卸作业;
- 5) 集装箱拖运车的放行检查;
- 6) 集装箱箱主对集装箱自然损坏维修费用的认同或请求集装箱堆场对损坏严重的集装箱做出报废的处理方法;
- 7) 集装箱的维修作业。

集装箱存取作业流程简单描述如下: 集装箱拖运车(以下简称集装箱拖车)来到堆场门口(又称闸口), 首先在排队区等候, 待集装箱检验、EIR 登记完毕(主要是进出场 EIR 输入、收费, 其中进场作业还要进行验箱, 即对进场空箱的完好状态进行验证), 根据进出场 EIR 的属性生成进出场作业单, 即指定集装箱存取堆位, 集装箱拖车按照进出场作业单所指定堆位驶至该堆位的附近等待搬运机服务, 待搬运机装卸作业完毕, 集装箱拖车即可离开堆场(其中, 取箱作业的集装箱拖车到堆场出口处集装箱拖车司机要进行验箱, 以确定所取箱符合出场 EIR 要求)。

在集装箱堆场管理中, 搬运机的作业调度直接反应了堆场作业成本的高低, 而等候装卸作业的集装箱拖车在堆场中的滞留时间是反应堆场客户服务水平的一个重要指标。目前, 大多数堆场的作业成本一直也无法得到有效的降低, 搬运机的油耗成本和维护成本都偏高。而且由于搬运机的作业效率低下等问题导致了大多数堆场都不同程度的出现了在高峰作业阶段等待作业的集装箱拖车排成长龙, 一方面影响公司的服务形象, 降低了市场的竞争能力, 另一方面, 严重阻碍了公共交通的情况。实际操作中, 堆场管理人员也想出了各种办法来解决这个问题, 如: 提高进出场检验、登记的自动化水平, 增加堆场作业设备数量或购买作业效率更高、自动化程度更高的作业设备, 安装计算机管理信息系统软件以加强信息化管理水平, 增加临时人手, 改善操作流程等等。但这些都需大量的投资, 为了降低风险, 大部分集装箱堆场只是采用了较成熟的技术、设备和作业工艺, 而且由于各集装箱堆场信息化水平参差不齐, 这在很大程度上制约了堆场作业效率的进一步提高。有的企业虽然建立了全厂的计算机网络, 但应用仍是分散的, 没有实现信息的共享和资源的优化配置。

以深圳市鸿基物流的集装箱堆场管理模式为基本依据, 项目组开发一个基于

internet 网络的集装箱堆场管理信息系统, 从而帮助该公司实现了内部的信息共享, 针对堆场作业流程中一些瓶颈环节如搬运机作业效率低下, 运用了运筹优化方法重新优化搬运机作业路线, 从而提高了作业效率, 实现了资源的优化配置。

该公司过去的搬运机作业调度是采用了先到先服务的原则, 这种运作方式往往导致了作业设备的空驶率很高, 几乎达到了 50%, 而且又直接导致了集装箱拖车等待时间的延长。假设系统随机产生 20 个作业点, 运用先到先服务的原则搬运机作业路线见图 3.1。从图中我们可以看出搬运机在堆场来回空跑, 既增加了油耗, 又降低了作业效率, 而且实验结果表明: 搬运机作业完这 20 个作业点所行驶的路径总长度为 2268 米, 集装箱拖车总等待时间为 991 分钟。

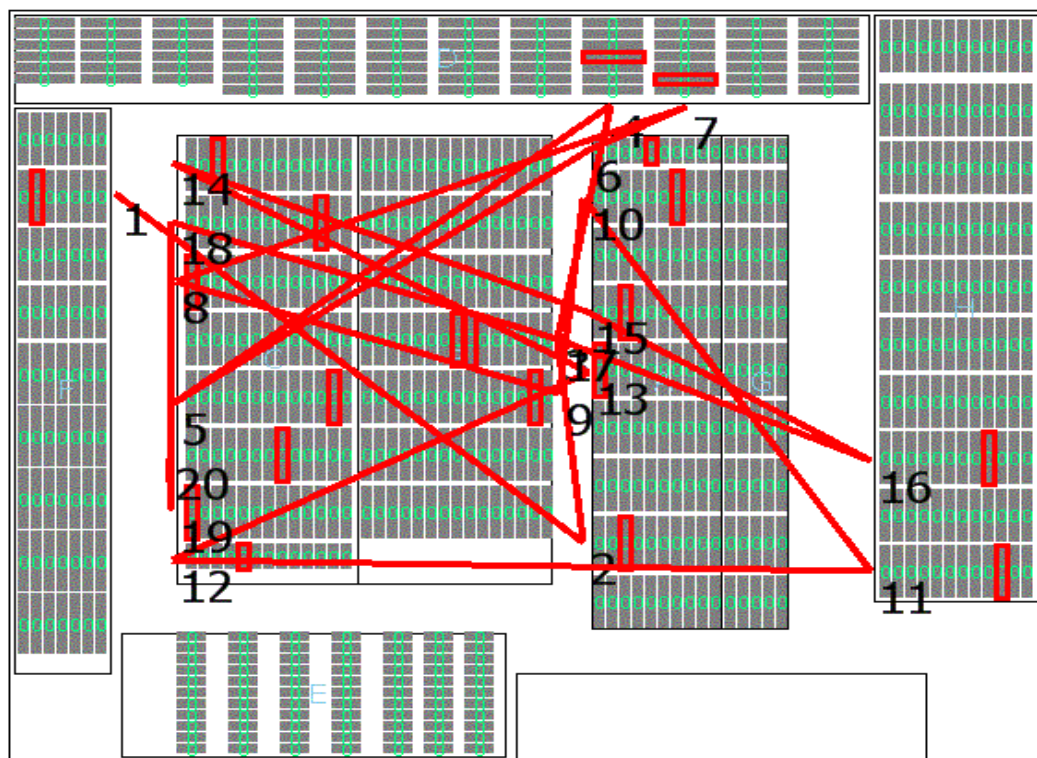


图 3.1 采用先到先服务的原则生成的搬运机作业路线图

Figure 3.1 Crane operation rout based in FIFO (First In First Out) method

3.2 集装箱堆场运作管理的模型建立

根据以上的问题, 本文将该问题用数学模型转化为 TSP 问题。

首先, 针对堆场作业做出以下几个假设:

(1) 只考虑单搬运机作业的情况。实际上, 即使有多部搬运机作业, 通常可以将整个堆场划分为几个小区域, 每部搬运机负责一个指定区域的作业, 这样一方面可以避免搬运机作业过程中的冲突, 另一方面, 便于作业的安排和搬运机作业效率的有效发挥。于是, 多搬运机问题就可转化为单搬运机问题。因此, 本文只对单搬运机作业问题进行分析。

(2) 每个堆列对应堆场路径中一个节点, 然后由这些节点组成整个堆场的路径图, 见图 3.2; 不同装卸操作指令如果位于同一堆列的对应于同一节点。

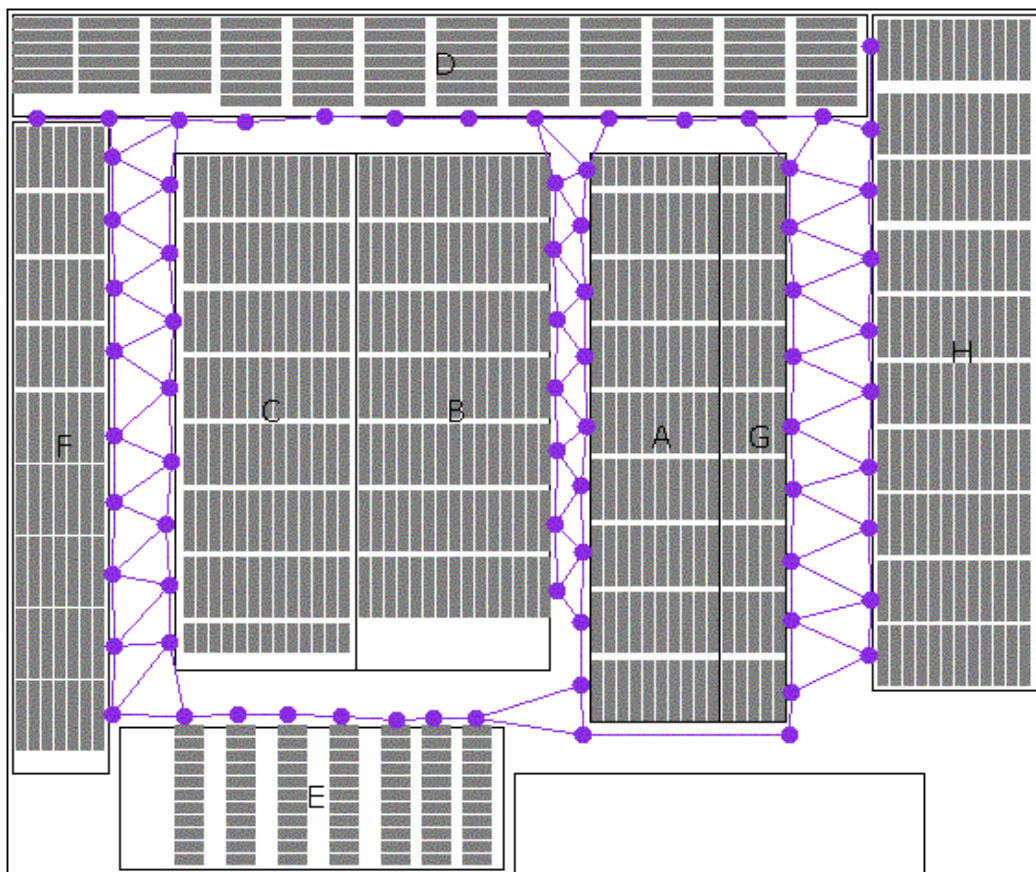


图 3.2 堆场路径图

Figure 3.2 Container yard rout

(3) 每辆集装箱拖车的装卸时间已知，且与接受服务的次序无关。由于在装箱作业时，如果该箱子不在堆列（yard-bay）的最外面堆（stack）的最上层，就要进行倒箱作业。这样，有可能就会影响在该堆列排在后面的等待作业的集装箱拖车，它们的装卸时间有可能取决于作业的次序。在这种情况下，排序问题就变成了另一个问题。所以，本文假设即使不同集装箱拖车的装卸作业时间有所不同，它们的装卸时间不受作业次序的影响。

(4) 每辆集装箱拖车都有各自的接受服务的期望时间。如果服务时间超过期望时间，则产生一延迟惩罚。

(5) 假设集装箱拖车到达堆场就可以进入堆场等待作业，即集装箱拖车从入闸口登记结束到进入堆场并到达指定点位置的时间不计。

(6) 假设搬运机匀速行驶，即不计搬运机转向掉头等变速运动。

在集装箱堆场管理系统中，假设在搬运机要处理 n 个集装箱的操作指令。第 i 个集装箱的装卸操作所用时间为 p_i ，第 i 个集装箱在堆场内的实际流通时间为 F_i ，第 i 个集装箱的交货期为 d_i （假设为 30 或 20 分钟）， s_{ij} 为紧接着第 j 个集装箱后操作第 i 个集装箱的准备时间（即搬运机从集装箱 j 所存放点运行到集装箱 i 所存放点的运行时间）， $s_{ij} = c_{ij}/v$ ，其中 c_{ij} 表示 j 点到 i 点的路程长度， v

表示搬运机平均运行速度。第 i 个集装箱的延迟时间 $L_i = F_i - d_i$ ，第 i 个集装箱的延误时间 $T_i = \max\{F_i - d_i, 0\}$ 。在该问题中，引入排序变量 z_{ix} 。

当第 i 个集装箱操作排在第 x 位置时 $z_{ix} = 1$ ；

否则， $z_{ix} = 0$ 。

首先，考虑搬运机操作完所有集装箱所行驶的总里程，见式 3.1

$$S_1 = \sum_{j=1}^n \sum_{i=1}^n \sum_{k=2}^n c_{ij} z_{ik} z_{j,k-1} \quad (3.1)$$

其中 c_{ij} 表示 j 点到 i 点的路程长度。

然后考虑所有集装箱总的延误时间，见式 3.2

$$S_2 = \sum_{i=1}^n \sum_{x=1}^n L_x R_{ix} z_{ix} \quad (3.2)$$

其中， R_{ix} 为选择函数

当 $L_x > 0, R_{ix} = 1$;

当 $L_x \leq 0, R_{ix} = 0$

这里 L_x 表示第 x 位置上操作的延迟时间，见式 3.3

$$L_x = \begin{cases} \sum_{i=1}^n p_i z_{i1} - \sum_{i=1}^n d_i z_{i1}, & \text{当 } x=1 \\ \sum_{i=1}^n p_i z_{i1} + \sum_{j=1}^n \sum_{i=1}^n \sum_{k=2}^x (p_i + s_{ij}) z_{ik} z_{j,k-1} - \sum_{i=1}^n d_i z_{ix}, & \text{当 } x \geq 2 \end{cases} \quad (3.3)$$

在进行多目标决策时，要求若干个目标同时都实现最优往往是很难的。由于直接解决多目标决策较困难，因此我们可以通过线性加权将目标化多为一，最终转化为一单目标规划问题。本文研究多目标车辆路线安排问题(MVRP)。由于MVRP问题不同目标之间相互制约，使其建模和求解具有一定的难度。文中分别引入平均单位运输成本 C_1 和平均单位等待时间惩罚成本 C_2 将作业设备总行驶路程和各

集装箱拖车在场总等待时间分别转化为总运输成本和总时间惩罚成本，从而将作业设备总行驶路程目标和各集装箱拖车在场总等待时间目标统一转化为总成本函数，实现了将原多目标问题转化为单目标问题求解。

这样，该问题（单机多目标调度问题）的目标函数的模型可以描述为式 3.4

$$S = C_1 S_1 + C_2 S_2 \quad (3.4)$$

式中， C_1 为平均单位运输成本， C_2 为平均单位时间惩罚成本。

上式的最小化将产生获得满足多目标的调度决策变量 z_{ix} ，当然还要满足约束条件，见式 3.5:

$$s.t. \begin{cases} \sum_{x=1}^n z_{ix} = 1, i=1, 2, \dots, n \\ \sum_{i=1}^n z_{ix} = 1, x=1, 2, \dots, n \end{cases} \quad (3.5)$$

一个有价值，有效率并且效率高、空驶率低的搬运机作业次序体现了集装箱拖车司机的满意度和低作业成本两方面的均衡。确定排序的原则是既要保证足够的服务质量（即减少总延误时间），又要保证搬运机的作业成本尽量小（即减少搬运机运行路程）。

3.3 实用的 TSP 算法——模拟退火法

由于组合优化问题在工程乃至经济领域有着广泛的应用，故如何获得组合优化问题的最优解，一直是人们试图解决的问题。不幸的是，许多实际问题都是 NP-完全的。严格求解这类问题所需的计算费用将是问题规模的指数形式。作为一种通用的随机搜索算法，模拟退火算法 SA (Simulated Annealing) 有着较好的渐近行为。该算法的思想是 1953 年 Metropolis N. 等人在研究二维相变时提出的，SA 算法最早分别由 Kirkpatrick S. 等人(1980)和 Cerny V. (1981) 独立地提出。

以下介绍的内容主要参考文献为[47][48][49]。

3.3.1 模拟退火算法的特点

SA 算法在搜索策略上与传统的随机搜索方法不同，它不仅引入了适当的随机因素，而且还引入了物理系统退火过程的自然机理。这种自然机理的引入使模拟退火算法在迭代过程中不仅接受使目标函数变“好”的试探点，而且还能以一定的概率接受使目标函数值变“差”的试探点，接受概率随着温度的下降而逐渐减小。SA 算法的这种搜索策略有利于避免搜索过程因陷入局部最优解而无法自拔的弊端，有利于提高求得全局最优解的可靠性。SA 算法的上述特性不仅在理论上能突破传统算法难以解决的难题，而且具有很强的科学和实际的工程应用价值，因而被誉为解决许多高难度优化问题的救星。下面分析 SA 算法和其它传统搜索方法的对比：

解析法是常用的搜索方法之一。它通常是通过求解使目标函数梯度为零的一组非线性方程来进行搜索的。一般而言，若目标函数连续可微，解的空间方程比较简单，解析法还是可以用的。但是若方程的变量有几十或几百时，它就无能为力了。爬山法也是常用的搜索方法，它和解析法一样都是属于寻找局部最优解的方法。对于爬山法，只有在更好的解位于当前解附近的前提下，才能继续向优解搜索。显然这种方法对于具有单峰分布性质的解空间才能进行行之有效的搜索，并得到最优解。

另一种典型的搜索方法是穷举法。该方法简单易行，即在一个连续有限搜索空间或离散无限搜索空间中，计算空间中每个点的目标函数，且每次计算一次。

显然,这种方法效率太低而鲁棒性不强。许多实际问题所对应的搜索空间都很大,不允许一点一点地慢慢求解。

随机搜索方法比起上述搜索方法有所改进,是一种常用的方法,但它的搜索效率依然不高。一般而言,只有解在搜索空间中形成紧至分布时,它的搜索才有效。但这一条件在实际应用中难于满足。这里必须把随机搜索(Random Search)方法和随机化技术(Randomized Technique)区分开来。SA方法也是利用随机化技术来指导对于最小能量状态的搜索。而另一个搜索方法——遗传算法就是一个利用随机化技术来指导对一个被编码的参数空间进行高效搜索的方法。因此,随机化搜索技术并不意味着无方向搜索,这一点与随机搜索有所不同的。

前述的几种传统的搜索方法虽然鲁棒性不强,但这些方法在一定的条件下,尤其是将它们混合使用也是有效的。当面临更为复杂的问题时,必须采用像SA算法这样更好的方法。SA算法具有十分顽强的鲁棒性,这是因为比起普通的优化搜索方法,它采用许多独特的方法和技术。主要有以下几个方面:

1) 以一定的概率接受恶化解

SA算法在搜索策略上与传统的随机搜索方法不同,它不仅引入了适当的随机因素,而且还引入了物理系统退火过程的自然机理。这种自然机理的引入使模拟退火算法在迭代过程中不仅接受使目标函数变“好”的试探点,而且还能以一定的概率接受使目标函数值变“差”的试探点,迭代中出现的状态是随机产生的,并且不强求后一个状态一定优于前一个状态,即以一定的可能容忍的退化状态的出现。接受概率随着温度的下降而逐渐减小。传统的方法往往是从解空间一个初始点开始最优解的迭代搜索过程。如登山法,若一个细微变动能改善质量,则沿该方向前进,否则取相反方向。然而复杂问题会使解空间中出现若干局部最优解,传统的方法很容易限于局部最优解而停滞不前。很多传统的优化算法往往是确定性的。从一个搜索点到另一个搜索点的转移有确定的转移方法和转移关系,这种确定性往往可能使得搜索永远达不到最优点,因而限制了算法的应用范围。而SA算法是以一种概率的方式来进行的,从而增加了其搜索过程的灵活性。

2) 引进算法控制参数 T

引进类似于退火温度的算法控制参数 T ,它将优化过程分成各个阶段,并决定各个阶段下随机状态的取舍标准,接受函数由Metropolis算法给出一个简单的数学模型。SA算法的两个重要步骤是:一是在每个控制参数 T 下,由前迭代点 $x(i)$ 出发,产生邻近的随机状态 $x(i+1)$,由 T 确定的接受准则决定此新状态的取舍,并由此形成一定长度的随机Markov链;二是缓慢降低控制参数 T ,提高接收准则,直至 $T \rightarrow 0$,状态链稳定于优化问题的最优状态,提高SA算法获得全局最优解的可靠性。

3) 使用对象函数值(即适应值)进行搜索

传统搜索算法不仅需要利用目标函数值,而且往往需要目标函数的导数值等其它一些辅助信息才能确定搜索方向。当这些信息不存在时,算法就无效了。而SA算法仅使用由目标函数变换来的适应度函数值,就可确定进一步的搜索方向和搜索范围,无需其它一些辅助信息。需要着重提出的是,SA算法的适应度函数不仅不受连续可微的约束,而且其定义域可以任意设定。对适应度函数唯一要求是,对于输入可计算出加以比较的正的输出。这个特性对很多无法或很难求导数的函数,或导数不存在的函数的优化问题,以及组合优化问题等,应用SA算法就显得比较方便。另外,直接利用目标函数值或个体适应度,也可以把搜索范围集中到适应度较高的部分搜索空间中,从而提高了搜索效率。

4) 隐含并行性

并行算法是60年代发展起来的, 发展迅速。有些专家甚至认为提高目前计算机系统性能的唯一方法是“选择大量的并行”。从目前情况看, 并行算法的设计主要采用两种方法: 一是对现有的串行算法加工改造, 使之成为好的并行算法; 二是结合所用并行计算机的结构特点, 直接设计新的并行算法。对模拟退火算法改造为并行算法还是比较容易的。目前常见的有以下几种并行策略: 操作并行策略, 试演并行策略, 区域分裂策略, 混乱松弛策略。这几种并行算法在不同程度上对解的质量、收敛速度方面较模拟退火算法优。由此可以预见, 大规模的并行计算模式将成为研究全局优化问题的主流。即SA算法隐含并行性(Implicit Parallelism), 它是优于其它求解过程的关键所在。另外SA算法的隐含并行性还有助于处理非线性问题。

5) 搜索复杂区域

SA算法最善于搜索复杂地区, 从中找出期望值高的区域。但在求解简单问题时效率并不高。正如遗传算法创始人Holland H.所指出的“如果只对几个变量作微小的改动就能进一步改进解, 则最好使用一些更普通的方法, 来为遗传算法助一臂之力”。SA算法在这一点上与遗传算法类似, 但比遗传算法更加适合搜索复杂区域。

上述具有特色的技术和方法使得SA算法使用简单、鲁棒性强、易于并行化, 从而应用范围甚广。

3.3.2 模拟退火算法的物理基础

模拟退火算法是 80 年代发展起来的一种用于求解大规模优化问题的随机搜索算法, 它以优化问题求解过程与物理系统退火过程之间的相似性为基础: 优化的目标函数相当于金属的内能; 优化问题的自变量组合状态空间相当于金属的内能状态空间; 问题的求解过程就是找一个组合状态, 使目标函数值最小。利用 Metropolis 准则并适当地控制温度的下降过程实现模拟退火, 从而达到求解全局优化问题的目的。本节首先介绍 SA 算法的物理基础。

物理系统退火过程

物理系统的退火(固体的退火)是一个先将固体加热至熔化, 再徐徐冷却使之凝固成规整晶体的热力学过程, 属于热力学与统计物理研究的范畴。

当加热固体时, 固体粒子的热运动不断增强, 随着温度的升高, 粒子与其平衡位置的偏差越来越大。当温度升至熔点以后, 固体的规则性被彻底破坏, 固体融解为液体, 粒子排列从较有序的结晶态转变为无序的液态, 这个融解过程的目的是清除系统中原先可能存在的非均匀状态, 使随后进行的冷却过程以某一平衡态为起始点。融解过程与系统的熵增过程相联系, 系统的能量也随温度升高而增大。

冷却时, 液体粒子的热运动逐渐减弱, 随着温度的慢慢降低, 粒子运动趋向于有序。当温度降至结晶点以后, 粒子运动变成围绕着晶体结点的微小振动, 液体凝固成固体的结晶态, 这个过程就是退火过程。退火过程之所以必须“慢慢”进行, 是为了使该物理系统在每一温度下都达到平衡态, 并最终达到固体的基态。在这个过程中, 系统的熵值不断减小, 系统的能量也随温度的降低趋于最小值。但是冷却时急剧降低温度, 会引起淬火效应, 即固体只能冷凝为非均匀。

退火过程中系统在每一温度下达到平衡态的过程,可以用封闭系统的等温过程来描述。根据 Boltzmann 有序性原理,退火过程遵循热平衡封闭系统的热力学定律——自由能减小定律:对于与周围环境热量而温度保持不变的封闭系统,系统状态的自然变化总是朝着自由能减小的方向进行,当自由能减小到最小值时,系统达到平衡态。根据该定律,能量减小与熵增加有利于自发变化,因此任一恒定温度下,系统状态从非平衡态自发转变到平衡态,都是能量和熵竞争的结果,温度决定着这两个因素的相对权重。在高温下,熵占统治地位,有利于变化的方向就是熵增加的方向,因而显出粒子的无序状态;而低温对应于低熵,低温下能量占优势,能量减小的方向有利于自发变化,因而得到有序(低熵)和低能量的晶体结构。

二、Metropolis 准则

固体在恒定温度下达到热平衡的过程可以用 Monte-Carlo 方法进行模拟,但由于必须采用大量采样才能得到比较精确的结果,因此计算量很大。从物理系统倾向于能量较低的状态,而热运动又妨碍它准确落入最低状态的物理现象,采样时着重提取那些有重要贡献的状态,则可以较快地得到较好的结果。

1953 年 Metropolis 等提出重要性采样法,即用下述方法产生固体的状态序列:先给定粒子相对位置表征的初始状态 i ,作为固体的当前状态,该状态的能量是 E_i ;如果使随机选取的某个粒子的位移随机地产生一微小变化,得到一个新状态 j ,它的能量是 E_j ;如果 $E_j < E_i$,则该新状态就作为重要状态,否则考虑到热运动的影响,根据固体处于该状态的几率来判断它是否是重要状态,固体处于状态 i 和状态 j 的几率的比值等于相应的 Boltzmann 因子的比值,见式 3.6

$$P(t) = \{E_i \rightarrow E_j\} = \frac{1}{z(t)} \exp\left(-\frac{E_j - E_i}{k_B t}\right) \quad (3.6)$$

其中, $P(t)$ 是在温度 t 处的粒子处于内能 E_i 的概率分布函数, k_B 为 Boltzmann

常数, $z(t) = \sum_i \exp\left(-\frac{E_i}{k_B t}\right)$ 被称为配分函数。 P 是一个小于 1 的数,用随机数发生

器产生一个 $[0,1]$ 区间的随机数 A ,若 $P < A$,则新状态 j 作为重要状态,就以 j 取代 i 成为当前状态,否则仍然以 i 作为当前状态;重复上述新状态的产生过程,在大量迁移后,系统趋向能量较低的平衡状态,固体状态的概率分布趋于 Gibbs 正则分布。

由式 3.6 可知,高温下可接受与当前状态的能量差较大的新状态为重要状态,而在低温下只能与当前状态的能量差较小的新状态为重要状态,当温度趋于零时,接受 $E_j > E_i$ 的新状态 j 的概率也为零。上述接受新状态的准则称为 Metropolis 接受准则,相应的算法称为 Metropolis 算法,这种算法的计算量显著减小。

通过对上述物理现象的模拟,即可以得到函数优化的 Metropolis 接受准则。设 $L(S,f)$ 为优化中的一个实例, S 表示解空间, $f: S \rightarrow R$ 表示解空间到实数域的映射, t 为 SA 过程中温度的控制参数。假定 $L(S,f)$ 存在着邻域以及相应解的产生

机制, $f(i)$, $f(j)$ 分别为对应于解 i , j 的目标函数值。由解 i 过渡到解 j 的接受概率用以下的 Metropolis 准则确定, 见式 3.7

$$P(t_k) = P(i \rightarrow j) = \begin{cases} 1, & f(i) \geq f(j) \\ \exp\left(\frac{f(i) - f(j)}{t_k}\right), & f(i) < f(j) \end{cases} \quad (3.7)$$

3.3.3 模拟退火算法的结构和数学模型

一、SA 算法的结构

SA 算法的执行策略由如下步骤构成: 从一个任意被选择的解开始探测整个空间, 并且通过扰动该解而产生一个新解, 按照 Metropolis 准则是否接受新解, 相应的下降控制温度, 基本的 SA 算法如下描述:

```
simulated-annealing()
{
    initialize ( $i_{start}$ ,  $t_0$ ,  $l_0$ );
    k=0 ;
    i= $i_{start}$  ;
    do
    {
        for(L=1; L<= $l_0$ ; L++)
        {
            generate(j,i);
            Metropolis (j,i);
        }
        k=k+1;
        update( $l_k$ ,  $t_k$ , k);
    }while stop-criterion()
}
```

在上述算法中 i_{start} , t_0 , l_0 分别表示初始状态的解, 控制参数(相当于温度 t)以及解产生次数的初始值。下标 k 表示第 k 轮的迭代。表示第 k 轮迭代中解产生的次数。函数 initialize (i_{start} , t_0 , l_0) 表示初始化, generate (j , i) 表示从解 i 产生一个新的解 j , Metropolis(j , i) 表示解的接受准则, update(l_k , t_k , k) 表示更新 l_k , t_k , k 的值, stop-criterion() 表示算法的收敛判断准则。

二、SA 算法的数学模型

在第 k 次迭代中被 SA 访问的解是 i , 而在第 $(k+1)$ 次迭代中被 SA 访问的解是 j 的概率: 它由两个独立的概率分布构成, 在第 k 次迭代中从解 i 产生解 j 的

概率 $g_{ij}(T)$ ，其中 $g_{ij}(T)$ 要求满足归一化条件： $\sum_{j \in \Omega_j} g_{ij}(T) = 1$ ，解被接受的概率

$a_{ij}(T)$ ，这里 T 是第 k 次迭代时的温度。对于 $i \neq j$ 的情况，转移概率的表达式见式 3.8

$$P_{ij}(T) = \{X_{k+1} = j | X_k = i\} = \begin{cases} g_{ij}(T)a_{ij}(T), \forall j \in \Omega_j \\ 0, \forall j \notin \Omega_j \end{cases} \quad (3.8)$$

因为 $a_{ij}(T)$ 不总是等于 1，故新解有不被接受的可能，算法停留在解 i 的概率为： $P_i(T) = 1 - \sum_{j \in \Omega} P_{ij}(T)$

由于 Ω 是一个可列集，故 SA 产生的随机变量所代表的随机过程是一个 Markov 链，其一步转移概率由式((2-3)和((2-4)定。记一步转移概率为：

$$P(T) = \begin{bmatrix} P_{11}(T) & P_{12}(T) & \cdots & P_{1|\Omega|}(T) \\ P_{21}(T) & P_{22}(T) & \cdots & P_{2|\Omega|}(T) \\ \vdots & \vdots & \ddots & \vdots \\ P_{|\Omega|1}(T) & P_{|\Omega|2}(T) & \cdots & P_{|\Omega||\Omega|}(T) \end{bmatrix} \quad (3.9)$$

则 k 步转移概率为：

$$P(m, m+k) = \begin{cases} \prod_{t=m}^{m+k-1} P(T_t), k \geq 1 \\ I, k = 0 \end{cases} \quad (3.10)$$

其中 I 为单位矩阵， T_t 表示第 t 次迭代时的温度值。

3.3.4 模拟退火算法的有限时间实现

一、概述

求解全局最优解是优化算法中的重要问题。当函数非凸或呈现分段连续时，难于利用传统的非线性规划方法求解全局最优解，因为传统的非线性规划方法是局部搜索算法。对于非光滑和高度病态的优化问题，以梯度为基础的非线性规划方法也往往无能为力。对于组合优化问题，传统的非线性规划方法效果也不好。近年来，人们提出了一些不依赖于具体问题的直接搜索方法，如 SA 算法、均场退火算法、禁忌搜索算法、遗传算法等软计算方法。SA 是近年来备受重视的一类软计算方法，能解决传统的非线性规划方法难于解决的某些问题，在 VLSI、生产调度、控制工程、机器学习、神经网络、图像处理、函数优化等许多领域得到广泛的研究。

SA 算法是一种随机性搜索方法，它以统计热力学为基础，引入了物理系统退火过程的自然机理。这种自然机理的引入不仅使模拟退火算法在迭代过程中接受一些较优的迭代点，而且还以一定的概率接受恶化解。它以随机过程中的有限状态奇异 Markov 链理论作为它的数学基础，接受概率随着控制温度的下降而逐

渐降低, 这种搜索策略能有效的使其避免陷入局部极小点。并且 SA 算法不需要任何的辅助信息(如梯度信息), 对目标函数和约束函数无任何要求, 具有较强的鲁棒性、全局收敛性、隐含并行性和广泛的适应性。SA 算法是按照自然法则计算的一个重要分支, 它将优化问题与统计力学中的热平衡问题相模拟, 另辟了求解全局优化问题的新途径。优化问题与金属退火过程在本质上是相似的: 优化问题的目标函数相当于金属的内能, 优化问题的变量组合状态空间相当于金属的内部状态空间, 优化问题的求解过程就是寻找一个组合状态使目标函数值最小。

SA 算法求全局优化问题时, 从理论上讲, 要对每个温度通过多次迭代达到平衡, 当温度从足够高降到足够低时, 就可以求出目标函数的最低点, 即全局最优解。对于理想模型, 经典模拟退火算法在下述条件下收敛于全局最优解: 初始温度足够高, 降温速度足够慢, 终止温度足够低。它表明算法的退火过程很慢, 计算效率不高。在全局优化问题中, 在理论上 SA 算法要对每个温度通过多次迭代达到平衡, 当温度从足够高降到足够低时, 就可以求出全局最优解。但是, 在每一温度下达到目标函数值的平衡, 需要的迭代次数是非常多的, 而且理想的退火过程要求温度连续下降, 这也是难以达到的。这些都是模拟退火算法效率不高的重要原因。由此, 初始温度的确定、解的产生机制、结束准则的确定、计算效率和计算精度的平衡, 都是构造模拟退火算法的重要问题。模拟退火算法的有限时间实现, 这在理论上和应用上都有十分重要的意义。

二、初始温度 T_0 的选取原则

尽管模拟退火算法在理论上是一种全局寻优算法, 不依赖于初始解的质量, 但是选取尽可能好的初始解可以提高算法效率, 减少运算时间。

由 Metropolis 准则 $\exp\left(-\frac{\Delta f}{T_0}\right) \approx 1$ 可知, T_0 应选取足够大, 以使算法一开始就达到准平衡, 否则退火过程将蜕变为一种局部随机搜索过程, 只能返回低质量的解。但是过大的 T_0 又可能导致过长的计算时间, 影响计算效率, 使 SA 算法失去可行性。

初始温度 T_0 的选取的一般方法是在 SA 算法开始以前, 观察系统在一系列干扰的情况下, 确定一个初始温度临界值。随机选定一个 T 值, 系统扰动 A 次, A_1 是使 $\Delta f_{ij} \leq 0$ 的次数, $A_2 = A - A_1$, 即 A_2 是使 $\Delta f_{ij} > 0$ 的次数, 若 A_2 次扰动被接受的次数是 B , 定义接受率 $K = B / A_2$ 。如果 K 接近 1, 则 $T_0 = T$; 否则按下式提高 T , $T = \max(\Delta f_{ij}) / (-\ln K)$, 重复试验直至 K 接近 1。求取临界温度需要花费很大计算量, 实际应用价值不大。而且临界温度方法还存在以下问题: (1) 临界温度的选取带有盲目性, 虽然有估计临界温度的经验公式, 但它不是普遍适用的。(2) 现有的 SA 算法经常采用类似于 Metropolis 算法的方法进行局部性的搜索, 因而对临界温度要求很高, 有时只能凭经验和具体问题获得, 影响了算法的通用性。由此产生数值计算估计和统计推断方法。

数值计算估计方法的基本思想是给出一个接近 1 的数 N 和初始温度 T , 按以下算法进行计算:

(1)初始化: $N, T, R_0=0, K=1$, Markov 链长 L , 一个常量 C ;

(2)在这个温度下迭代 L 步, 分别记录 SA 算法接受和拒绝的解的个数 X, Y , 并计算 $R_k = X / L$;

(3)当 $|R_k - N| < \epsilon$ 时, 停止计算; 当 $R_{K-1} < N$ 且 $R_k < N$ 时 $K=K+1, T=T+C$, 返回(2); 当 $R_{K-1} \geq N$ 且 $R_k \geq N$ 时 $K=K+1, T=T-C$, 返回(2); 当 $R_{K-1} \geq N$ 且 $R_k \leq N$ 时 $K=K+1, T=T+C/2, C=C/2$, 返回(2); 当 $R_{K-1} \leq N$ 且 $R_k \leq N$ 时 $K=K+1, T=T-C/2, C=C/2$, 返回(2);

通过上述算法, 可以估计出初始温度 T_0 .

三、邻域结构的迭代方法

模拟退火算法是依赖邻域结构的迭代方法, 如何找领域的解将直接影响收敛速度和最优解。针对这个问题, 提出 6 种策略, 这里假设有 n 个城市。

策略A 在第 1 ~ n 个访问的城市中随机地选取第 j_1 次和第 j_2 次访问的城市, 在路径 C_0 中交换第 j_1 次和第 j_2 次访问的城市, 其余不变, 此时的路径为 C_1 。

如: $C_0 = 2, 3, 4, 1, 5, 7, 9, 8, 6$. $j_1 = 2$ (第 2 次访问的城市是城市 3), $j_2 = 7$ (第 7 次访问的城市是城市 9), 则 $C_1 = 2, 9, 4, 1, 5, 7, 3, 8, 6$.

策略B 在第 1 ~ n 个访问的城市中随机地选取第 j_1 次访问的城市, 在路径 C_0 中交换第 j_1 次和第 $j_1 + 1$ 次访问的城市, 其余不变, 此时的路径为 C_1 。

如: $C_0 = 2, 3, 4, 1, 5, 7, 9, 8, 6$, $j_1 = 2$, 则 $C_1 = 2, 4, 3, 1, 5, 7, 9, 8, 6$.

策略C 也称逆转策略, 在第 1 ~ n 个访问的城市中随机地选取第 j_1 次和第 j_2 次访问的城市, 在路径 C_0 中第 j_1 次到第 j_2 次访问的城市之间的子路径以反方向插入, 其余不变, 此时的路径为 C_1 。

如: $C_0 = 2, 3, 4, 1, 5, 7, 9, 8, 6$, $j_1 = 2$, $j_2 = 7$, 则 $C_1 = 2, 9, 7, 5, 1, 4, 3, 8, 6$.

策略D 在第 1 ~ n 个访问的城市中随机地选取第 j_1 次和第 j_2 次访问的城市, 假设 $j_1 < j_2$, 在路径 C_0 中将第 j_1 次访问的城市安排到第 j_2 次访问的

城市之后,其余不变,此时的路径为 C_1 。

如: $C_0 = 2, 3, 4, 1, 5, 7, 9, 8, 6$. $j_1 = 2$, $j_2 = 7$, 则 $C_1 = 2, 4, 1, 5, 7, 9, 3, 8, 6$.

策略E 上面策略没有利用城市间距离大小的信息,策略E 将利用边的邻接关系,依据蚁群算法的思想,距离近的邻接点以较大的概率选为下一个访问点,所以在局部调整时依据此思想。

设 $d(i, j)$ 表示城市 i 与城市 j 的距离,在 $1 \sim n$ 的城市中随机地选取城市 i_1 ,离城市 i_1 最远的城市的距离为: $d_{\max} = \max d(i_1, j)$,为了排除下一个访问点为它自己,令 $d(i_1, i_1) = d_{\max}$,则下一个访问点的为城市 j 概率为

$$p_j = \frac{d_{\max} - d(i_1, j)}{\sum_{k=1}^n (d_{\max} - d(i_1, k))} \quad (3.11)$$

假设以上式的概率选取的是城市 j_1 , 在路径 C_0 中将城市 j_1 安排到城市 i_1 之后, 其余不变, 此时路径为 C_1 。

如: $C_0 = 2, 3, 4, 1, 5, 7, 9, 8, 6$. $i_1 = 3$, $j_1 = 7$, 则 $C_1 = 2, 3, 7, 4, 1, 5, 9, 8, 6$.

策略F 在策略E 中是在 $1 \sim n$ 的城市中随机地选取城市 i_1 , 为了使路径总长度和达到最短, 优先解决薄弱环节, 这里采用路径中相邻城市之间的距离大的两个城市以较大的概率被选取, 在它们之间插入其他城市。用 $l(n)$ 数组记录路径 C_0 相邻城市之间的距离, 具体数据如下:

$$l(k) = d[c(k), c(k+1)], k = 1, 2, \dots, n-1$$

$$l(n) = d[c(n), c(1)]$$

选取城市 i 的概率为:

$$p_i = \frac{l(i)}{\sum_{k=1}^n l(k)} \quad (3.12)$$

按上式以概率选取的是城市 i_1 , 后面方法同策略E, 按(1) 式以概率选取的是城市 j_1 , 在路径 C_0 中将城市 j_1 安排到城市 i_1 之后, 其余不变, 此时的路径为 C_1 。

四、Markov 链长 L_k 的选取原则

若在一定的温度下做无限次迭代, 则相应的 Markov 链可达到平稳分布概率。然而, 这在实际应用中是无法实现的, 现在的问题是: 如何合理设置 Markov 链

长 L_k ，以便在有限的时间内获得一个 Markov 链的拟平稳分布。

有文献指出 Markov 链长 L_k 满足 $L_k \geq C(|\Omega|^2 - 3|\Omega| + 3)$ 时，该 Markov 链基本上是平稳分布，其中 C 是一个和转移矩阵有关的参数。故按上式来设计 Markov 链长 L_k ，其计算时间将是问题的指数规模。

在实际计算中，可采用固定总迭代次数的方法，也可采用固定 Markov 链长的方法，并取每个 Markov 链的长度大于最大邻域，即 $L \geq \max_{j \in \Omega} |\Omega_j|$ 。

Markov 链的长度的选取还与温度控制参数 T_k 的下降密切相关， T_k 的缓慢下降可以避免过长的 Markov 链。在温度控制参数 T_k 的下降幅度已选定的情况下， L_k 应选取在温度控制参数的每一值上都能恢复准平衡。根据这一原则， L_k 的值不应选取过小，一般取 $L_k = 100N$ ，其中 N 为问题的规模。

一般有以下两种方法：

(1) 固定长度法：通常采用与邻域大小直接相关的规则，如 TSP 的两城市位置所定义的邻域，它的大小是 $n(n-1)/2$ ，在同一温度下，它的迭代步长可取 $n, n/2, n^2, n^2/2, n^k (k>2)$

(2) 由接受和拒绝的比率来控制迭代步数：当温度很高时，每一个状态被接受的频率基本相同，而且几乎所有的状态被接受，此时，在同一温度应使迭代的步数尽量小。当温度渐渐变低时，越来越多的状态被拒绝，如果在此温度的迭代太少，则可能造成过早地陷入局部最优状态。比较直观和有效的方法是随着温度的下降，将同一温度的迭代步长增加。实现的一种方法是给定一个充分大的步长上限 U 和一个接受次数指标 R ，当接受次数等于 R 时，在此温度不再迭代而使温度下降，否则，一直迭代到上限步数。实现的第二种方法是给定一个接受比率指标 R 、迭代步长上限 U 和下限 L ，每一温度至少迭代 L 步且记录同一温度迭代的总次数的比率不小于 R 时，在这一温度不再迭代而开始温度下降，否则，一直迭代到上限步数。

五、温度下降函数的选取原则

为避免算法进程产生过长的 Markov 链，温度控制参数 T_k 的下降幅度应该以小为宜，这也是温度控制参数 T_k 的下降函数的选取原则。常用的温度控制参数 T_k 的下降函数为 $T_{k+1} = aT_k$ ，其中 a 是一个接近 1 的数，一般取作 0.5~0.99。这个温度控制参数 T_k 的下降函数具有随算法进程递减的下降量，因此可以减小温度控制参数 T_k 的递减速率，从而延缓了变换的接受概率随算法进程递减的态势，有利于 SA 算法性能的稳定。

在 SA 退火算法中, 降温的方式直接影响解的产生方式、接受方式和结束方式, 因此对算法的计算效率和适用性影响很大。温度下降太快, 可能会丢失极值点; 温度下降过慢, 算法的收敛速度又大为降低。为了提高 SA 算法的计算效率和适用性, 许多学者提出了多种降温方案, 有代表性的是:

(1)经典退火方式: 降温公式为 $T(t) = T_0 / \ln(1+t)$, 特点为温度下降速度很缓慢, 因此算法的效率很低;

(2)快速退火方式(FSA):降温公式 $T(t) = T_0 / (1+at)$, 这种退火方式的特点为在高温区, 温度的下降比较快;在低温区温度的下降比较慢。这符合热力学分子运动理论, 某粒子在高温时具有较低能量的概率要比低温时小得多, 因此寻优的重点应在低温区。式中 a 可以改善退火曲线的形状;

(3)超快速退火方式 (VFSA):降温公式为 $T(K) = T_0 \exp(-CK^{1/N})$, 其中 K 为迭代次数, C 为给定常数, N 为变量的个数。它采用依赖于温度的似 Cauchy 分布产生新模型, 是 1989 年 Ingber 首先提出的。其解的产生方式见式 3.13, 3.14

$$m_i = m_i + y_i (B_i - A_i) \quad (3.13)$$

$$y_i = T \operatorname{sgn}(u - 0.5) \left[(1 + 1/T)^{|2u-1|} - 1 \right] \quad (3.14)$$

其中 m_i 为模型中第 i 个变量, u 为 [0, 1] 均匀分布的随机数, $[A_i, B_i]$ 为 m_i 的取值范围。这种退火方式的特点是在高温情况下进行大范围搜索, 在低温时仅在当前模型附近搜索。而且由于似 Cauchy 分布有一平坦的尾部, 使其易于跳出局部极值。这一改进大大加快了算法的收敛速度。

六、结束准则的选取原则

SA 算法从初始温度开始, 通过在每一温度的迭代和温度的下降, 最后达到终止准则而停止。尽管有些原则有一定的理论指导意义, 但大多数是直观的。下面分几类讨论如下:

(1)零度法: SA 算法的最终温度为零, 因而最为简单的原则是: 给出一个较小的正数, 当温度小于这个数时, 算法停止, 表示已经达到最低温度。

(2)循环总数控制法: 总的下降次数为一定值, 当温度迭代次数迭代此数时, 算法停止。

(3)基于不改进规则的控制法: 在一个温度和给定的迭代次数内没有改进当前的局部最优解, 算法停止运算。SA 算法的一个基本思想是跳出局部最优解, 直观的结论是在较高的温度下没能跳出局部最优解, 则在低的温度下跳出局部最优解的可能性也比较小。

(4)接受概率控制法: 给出一个较小的正数, 除当前局部最优解以外, 其它状态的接受概率都小于此数, 算法停止运算。

理论上是用一个 Markov 链描述 SA 算法的变化过程, 因此具有全局最优性, 实际运用中的 SA 算法是一个启发性算法。它有诸多的参数需要调整, 即 SA 算法的冷却方案是指一个控制参数值的有限序列: 初始温度 T_0 、由降温准则确定的温度衰减函数、由结束准则确定的控制参数的终值。SA 算法要解决实际优化问

题，必须能在有限的时间内实现，为此必须提供一个比较通用而又高效的冷却方案。但目前难以找到高效通用的冷却方案，只是根据具体应用问题和有关经验来确定冷却方案，当问题规模扩大时这是不可行的。

模拟退火算法的算法流程见图 3.3

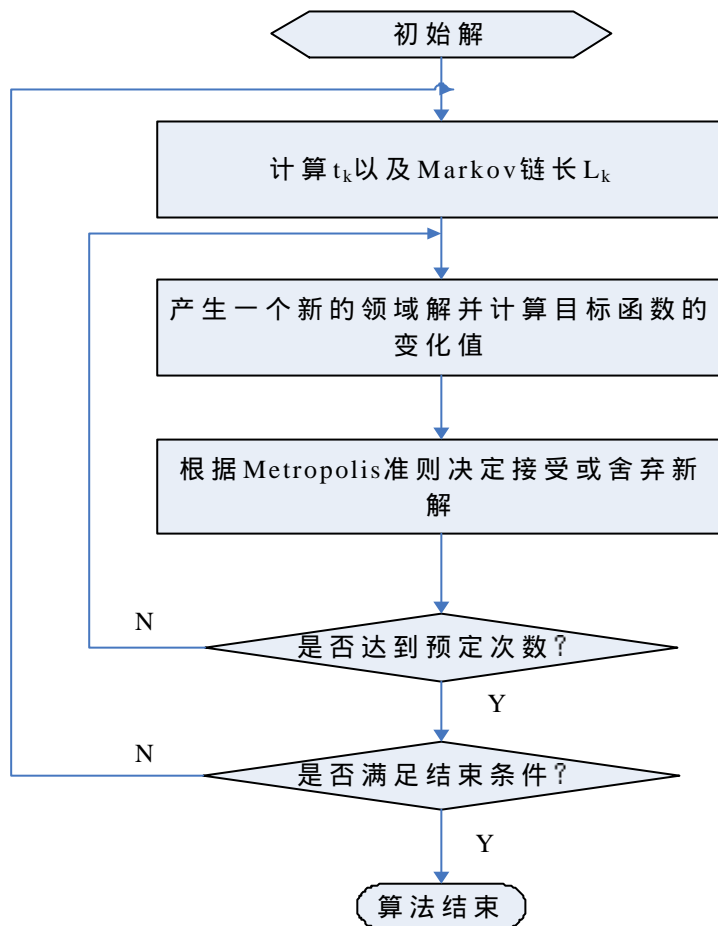


图 3.3 模拟退火算法的流程框图

Figure 3.3 Simulated Annealing flow chart

我们针对在本章第一节中模拟的 20 个作业点，运用模拟退火算法得出的搬运机作业路线见图 3.4。相比图 3.1 运用先到先服务的原则作业的搬运机作业路线图来说，搬运机的空驶率得到了大大的降低，大大提高了作业效率，而且实验结果表明：搬运机作业完这 20 个作业点所行驶的路径总长度为 793 米，所有集装箱拖车的总的等待时间为 858 分钟，相比先到先服务的作业方法来说，搬运机所行驶的路径大大减少了，同时集装箱拖车的等待时间也得到了降低。

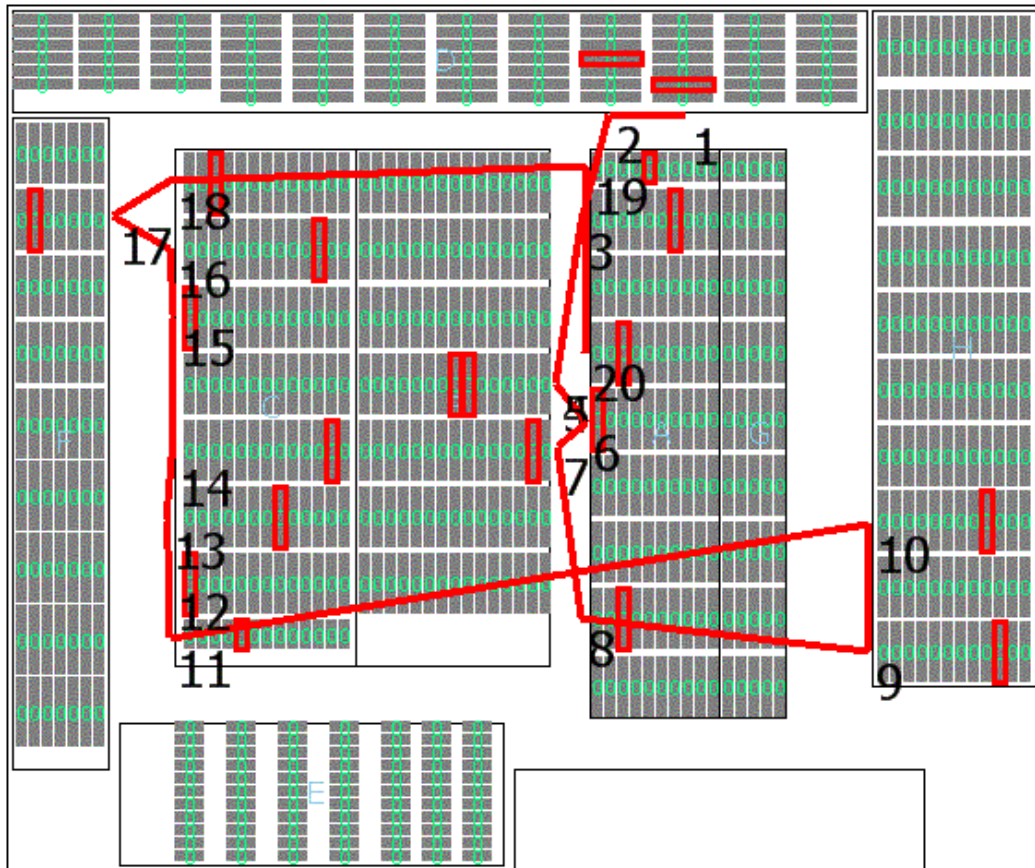


图 3.4 运用模拟退火算法得出的搬运机作业路线图

Figure3.4 Crane operation rout based in Simulated Annealing

第四章 基于模拟退火算法的集装箱堆场作业排序优化计算

4.1 集装箱堆场实际运行参数

由于堆场作业点分布是确定的，即搬运机所行驶路径节点是有限而且固定的，因此为了缩短算法计算时间，本文提出将本文所研究的搬运机作业路径优化问题分成两部分：第一部分通过最短路径算法计算出每两节点之间的最短路径走法和长度值；第二部分运用模拟退火算法求解多作业点的执行次序问题。

首先，根据堆场实际布置在系统中画出堆场布局图，见图 4.1。然后通过设置节点生成堆场路径图，见图 4.2，即假设堆场搬运机只沿着图中的路线行驶。

每相邻两节点的距离都计算出来并存入数据库中，见表 4.1，其中 ID 表示每条路径的唯一标识，Point1 ID 和 Point2 ID 字段表示这条路径的两端的节点号，Length 字段表示这条路径的长度。

然后运用最短路径算法算出任意两点之间的最短路径并存入数据库中，见表 4.2，其中 Point1 ID 和 Point2 ID 字段表示任意两节点号，Length 字段表示这两节点之间的最短路径长度，Path 字段表示这两节点之间的最短路径走法。如：节点 580 到节点 635 之间的最短路线为 580-581-644-643-642-641-648-649-650-651-652-635，表示 580 节点先后经过 581 节点，644 节点，643 节点，…，652 节点到达 635 节点这条路径为从 580 节点到 635 节点的最短路线，且路径长度为 127 米。

表 4.1 相邻两节点路径长度表

ID	Point1ID	Point2ID	Length
788	580	581	13
789	581	582	13
790	582	583	12
791	583	584	15
792	584	585	13
793	585	586	14
...

表 4.2 堆场中任意两节点间的最短路径表

Point1ID	Point2ID	Length	Path
580	634	136	580-581-644-643-642-641-648-649-650-651-652-635-634
580	635	127	580-581-644-643-642-641-648-649-650-651-652-635
581	634	123	581-644-643-642-641-648-649-650-651-652-635-634
581	635	114	581-644-643-642-641-648-649-650-651-652-635
582	619	98	582-583-584-585-586-587-622-620-619
582	620	86	582-583-584-585-586-587-622-620
...

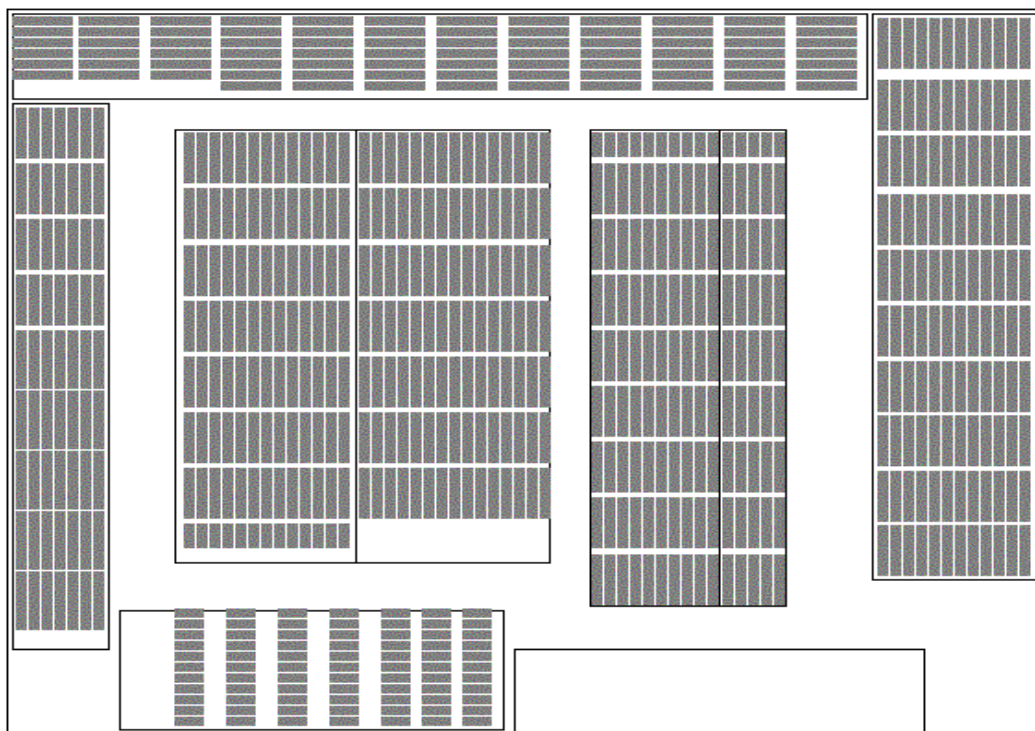


图 4.1 堆场布局图

Figure4.1 Container yard layout

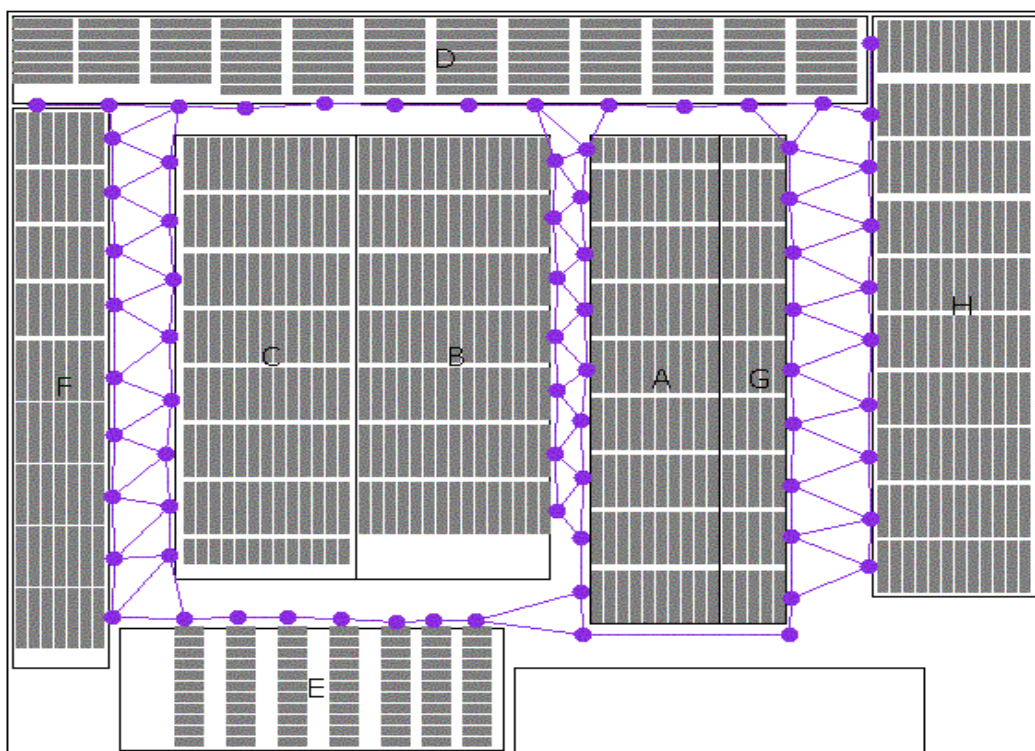


图 4.2 堆场路径图

Figure4.2 Rout in Container yard

4.2 模拟退火算法的参数设计

1. 初始温度的选取：初始温度的确定是模拟退火算法的一个难点，本文通过数值估计的方法进行试验研究。从理论上来说，初始温度 t_0 应保证对于任意的 i 和 j ，有 $\exp(-\Delta f_{ij}/t_0) \approx 1$ ，在该问题中，我们可以将 t_0 简单估计为： $t_0 = k\mathbf{d}$ ， k 为充分大的数，其中，

$$\mathbf{d} = \sum_{i=1}^n \max\{d_{ij} | j \neq i, j=1,2,\dots,n\} - \sum_{i=1}^n \min\{d_{ij} | j \neq i, j=1,2,\dots,n\},$$

在实际试验计算中，分别选取 $k=10, 20, 50, 100$ 等实验值进行了试验。

2. 解的产生机制和温度下降准则的选取：模拟退火算法成功与否的一个关键因素，是选取一个合理高效的降温准则，因为降温准则跟解的产生机制和结束准则有很大的关系。为避免算法进程产生过长的 Markov 链，温度控制参数 T_k 的下降幅度应该以小为宜，这也是温度控制参数 T_k 的下降函数的选取原则。本文选

取了超快速退火方式 (VFSA)：降温公式为 $T(K) = T_0 \exp(-CK^{1/N})$ ，其中 K 为迭代次数， C 为给定常数， N 为变量的个数。它采用依赖于温度的似 Cauchy 分布产生新模型，是 1989 年 Ingber 首先提出的。这种退火方式的特点是在高温情况下进行大范围搜索，在低温时仅在当前模型附近搜索。而且由于似 Cauchy 分布有一平坦的尾部，使其易于跳出局部极值，而且大大加快了算法的收敛速度。

解的产生机制即在同一温度下邻域结构的迭代方法，本文分别运用第三章中提出的几种退火策略针对该问题 20 个作业点做测试比较。首先假设算法的其他参数相同，起始温度 $T=5000$ ，终止温度 $T_s=0.05$ ，各种策略随机测试 50 次后，结果见表 4.3。

表 4.3 几种不同邻域结构的迭代方法应用的结果比较

算法	平均计算时间(s)	平均路程值 (m)	最好路程值 (m)	最差路程值 (m)
策略 A	11	872	825	967
策略 B	56	1003	947	1600
策略 C	18	780	767	812
策略 D	8	953	1121	812
策略 E	17	816	769	864
策略 F	20	786	769	837

从表 4.3 可以看出，策略 C 是最好的退火策略；而策略 A 和策略 D 花费的时间虽然少，但结果较差，说明它们很容易落入局部最优解；策略 B 效果最差，需要的时间也多。其中，策略 C 采用逆转策略，可以使迭代过程突破局部最优圈而跳到另一个搜索空间，而且无论从计算时间和解的值来说都是较优的，所以本文采用了策略 C 作为退火策略。

3. 解的接受准则和结束准则：

解的接受准则按照Metropolis算法来计算,在解的接受准则中还要处理奇异解问题:目标函数或约束函数中若有一个大于一个给定的足够大的正数,此解即为奇异解。遇到奇异解时,重新产生一个新解进行迭代,以保证算法的健壮性。

求解过程的结束准则,是模拟退火算法中最难处理的一个环节,其条件的严格或宽松决定了计算效率和求解精度。本文综合考虑迭代次数、当前可行最优解的稳定情况的方法,即为了保证算法的健壮性,本文设置了算法结束的温度

$t_s=0.05$,另外还设置了最大的降温迭代次数为50000次,超过此迭代次数后算法结束。

4.3 实验算例

首先根据现场调研数据和经验数据分析,我们假设参数设置如下:

每个集装箱的装卸时间设为 1.43 分钟;搬运机的速度设为每 3 秒钟经过一堆列;每辆集装箱拖车的期望服务时间为集装箱拖车进入堆场后 30 分钟,每单位时间的延迟惩罚成本设为 0.1 元/分钟,而搬运机单位距离运行成本为 0.12 元/km。

假设随机产生 10 个进箱作业点,并且两两进箱间隔时间按负指数分布,且根据现场调研数据,设其平均间隔时间为 2 分钟,作业点在堆场图中的分布见图 4.3,图中的加粗小方框为 10 个集装箱进场所摆放的堆位,序号表示集装箱的进场先后次序。

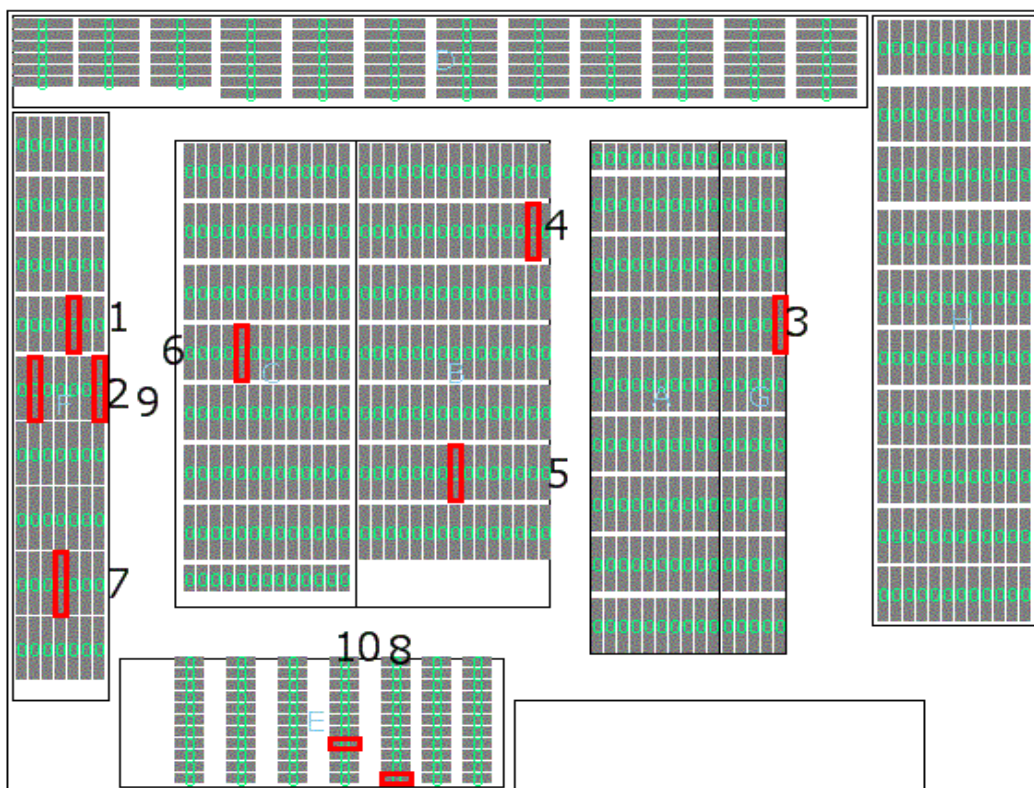


图 4.3 10 个作业点在堆场中的位置图

Figure4.3 Operation position in Container yard

目前大多数堆场实际操作中常用的搬运机调度方式是遵循先到先作业原则，运用这种原则对上图中 10 个点进行操作，可以得出搬运机作业路线图，见图 4.4。图中两点之间的连线并不代表搬运机所行驶的实际路径，搬运机行驶的实际路径是通过最短路算法得出的最短路径。而且通过计算可以得出，搬运机执行完所有作业所行驶的总路程为 1017，所有作业点的总等待时间为 182.6 分钟。从图中可以看出搬运机在场上来回跑，空驶率较高。

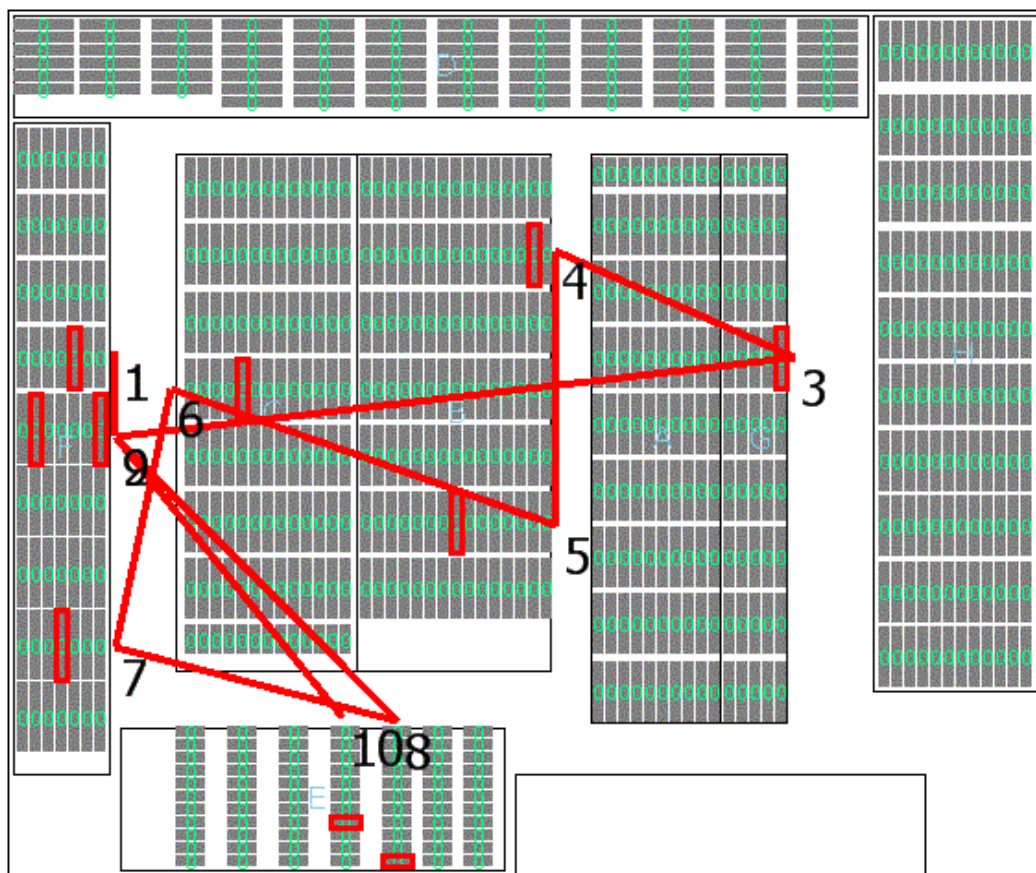


图 4.4 按先到先服务的原则得出的搬运机作业路线图

Figure 4.4 Operation route of crane based in FIFO

以下我们将运用模拟退火算法对搬运机的作业调度进行重新安排。

首先，由于本文所讨论的搬运机作业排序问题并不是简单的 TSP 问题，它是一个始点和终点不是同一点的问题，即搬运机完成所有的作业后并不需要返回起始点。所以，本文提出三种取终点的方法通过仿真实验进行比较，包括：选起点为最终点的算法、选最远点为最终点的算法、选最近点为最终点的算法。其中，选起点为最终点的算法即按照经典 TSP 问题从起点出发然后回到起点，然后从总路程中减去最后执行的点返回到起点的路程；选最远点为最终点的算法即选取距离搬运机当前所在位置最远的点为最好执行的作业点；选最近点为最终点的算法即选取距离搬运机当前所在位置最近的点为最好执行的作业点。

应用以上三种选取终点的方法分别做 50 次仿真实验，得出运用三种方法得出的结果，仿真实验结果见表 4.4。

表 4.4 选取最终点的方法仿真实验结果

算法	平均			最好		最差	
	路程值 (米)	计算时 间(秒)	总等待时 间(分钟)	路程值(米)	总等待时间 (分钟)	路程值(米)	总等待时间 (分钟)
选起始点 为最终点 的算法	420.8	3.5	153.8	402	151.6	492	156.5
选最远点 为最终点 的算法	478.8	2.9	155.2	440	152.2	586	159.6
选最近点 为最终点 的算法	538	2.9	157.9	528	152.2	555	160.8

从表 4.4 可以看出, 无论从搬运机所行驶总路径和集装箱拖车总等待时间来看, 选起始点为最终点的算法优于其他两种算法。因此, 本文在最终点的选取方法上选择了选起始点为最终点的算法。这三种方法得出的最好结果的作业路径图分别见图 4.5, 图 4.6, 图 4.7。图中, 1, 2, 3, ..., 表示搬运机对各堆位的集装箱作业的顺序, 各条连接线并不代表搬运机实际行驶路线, 搬运机实际行驶路线由两点之间的最短路确定。

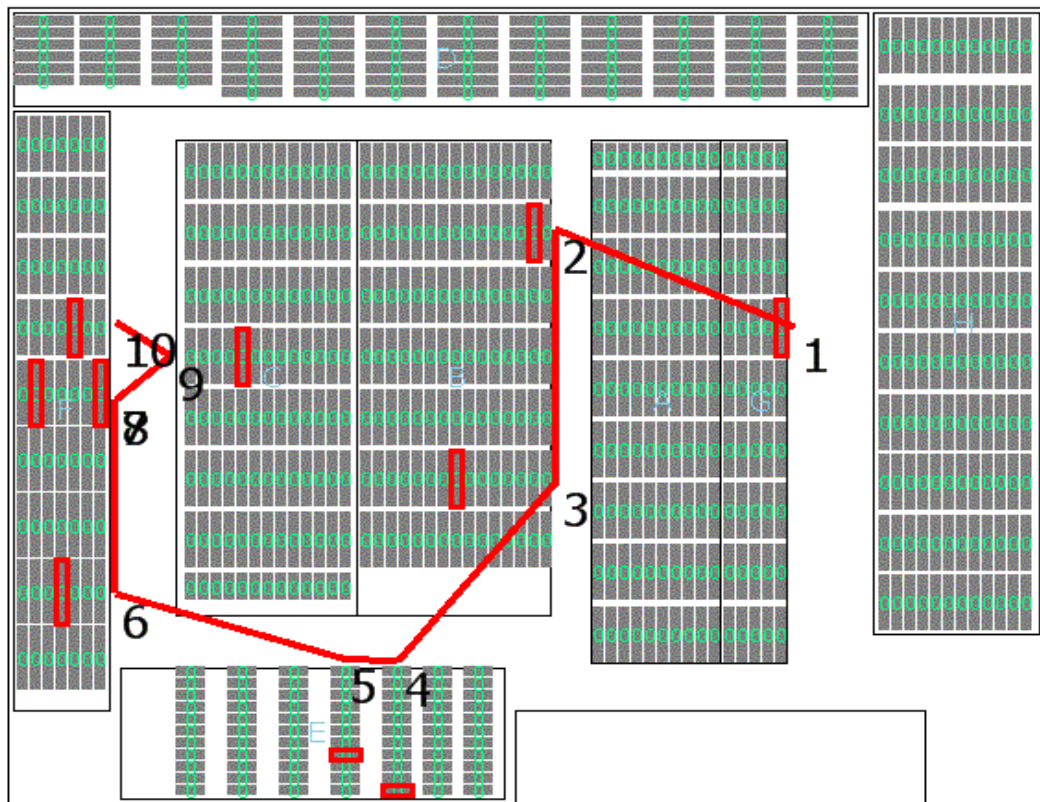


图 4.4 选起始点为最终点的算法得出的搬运机作业路线图

Figure 4.4 Crane operation route based in method which select begin point as last operation point

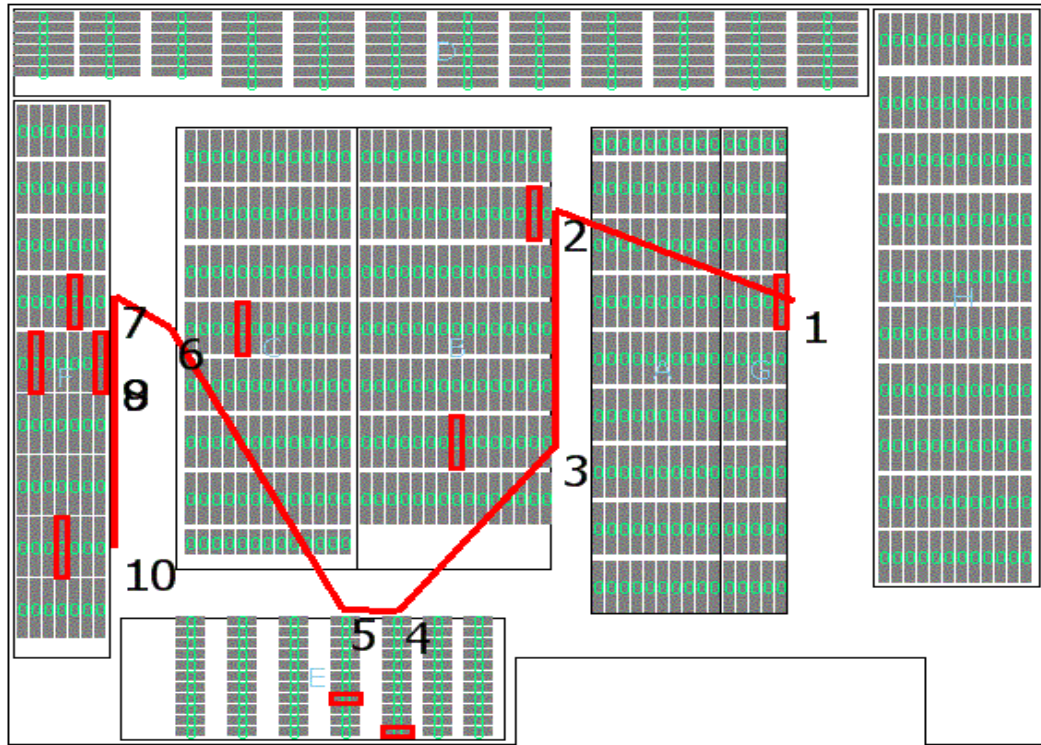


图 4.5 选最远点为最终点的算法得出的搬运机作业路线图

Figure4.5 Crane operation rout based in method which select farthest point as last operation point

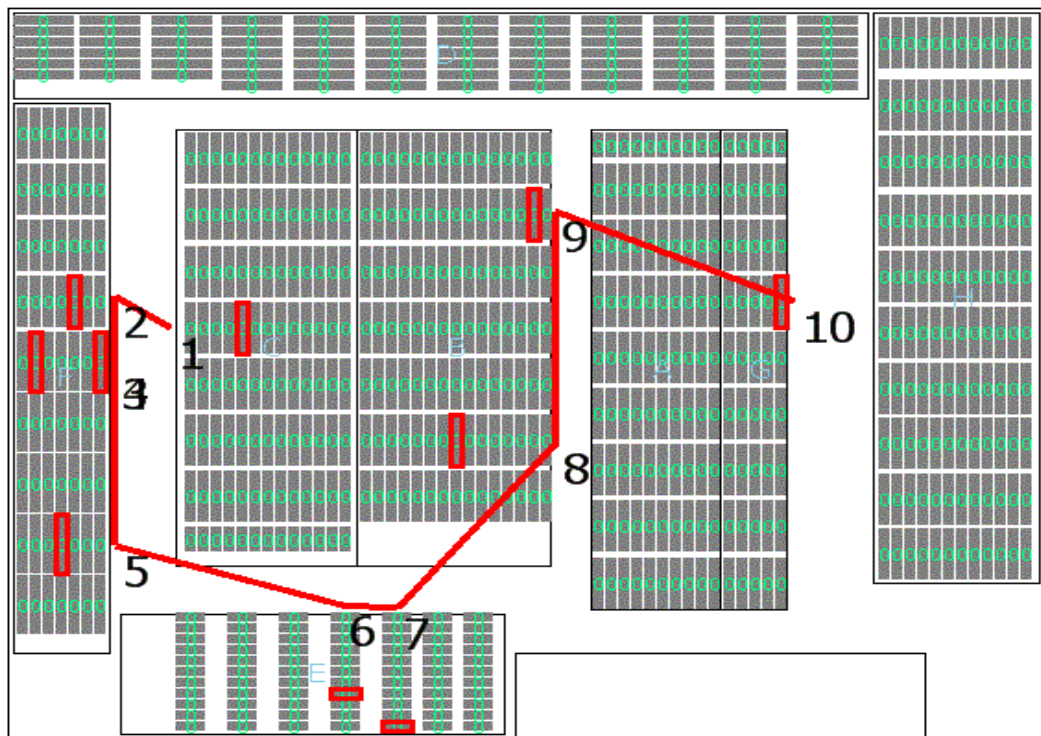


图 4.6 选最近点为最终点的算法得出的搬运机作业路线图

Figure4.5 Crane operation rout based in method which select nearest point as last operation point

通过以上的比较，可以看出运用模拟退火算法对搬运机作业调度重新排序，总等待时间得到了减少，而且搬运机的总行驶路程也得到了大大的减少，运用模拟退火算法和先到先服务的方法得出的结果见表 4.5。

表 4.5 模拟退火算法和先到先服务方法的结果比较

运用模拟退火算法	搬运机行驶总路程（米）	420.8
	集装箱拖车总等待时间（分钟）	153.8
采用先到先服务方法	搬运机行驶总路程（米）	1017
	集装箱拖车总等待时间（分钟）	182.6

从表 4.5 中，可以看出，搬运机运用模拟退火算法后，搬运机执行完 10 个作业点所行驶的总路程，相比于目前大部分堆场所采用的先到先服务的方法缩短了 58.6%，从搬运机运行成本来看，每公里的油耗成本为 0.12 元，则搬运机执行完 10 个点能节省约 0.075 元，根据现场调研，堆场一天进出的箱子数大概为 1000 个，则一天可以节省油费大约 75 元，则一年的油费可以节省大约 3 万元，这对于一个企业来说，成本得到了大大的降低。而且从表 4.5 中还可以看到运用模拟退火算法以后，集装箱拖车总等待时间也减少了，即顾客服务水平也因此得到了提高。

第五章 总结与展望

组合优化问题一直是管理科学领域的热门话题。许多组合优化问题尤其是其中的 NP 完备问题由于其固有的计算复杂性，求精确解的时间往往随问题规模的增大而呈指数型增长，甚至成为理论上可解而算法上无法实现的问题。因而，研究求组合优化问题近似解的算法亦得到了广泛的关注。

本文在全面介绍国内外求解组合优化问题算法和 TSP 以及 VRP 领域研究现状的基础上，提出了通过优化集装箱堆场搬运机作业调度，提高集装箱堆场的运营管理水平，降低其运营成本。以此为出发点，本文给出了基于搬运机行驶路程成本和集装箱拖车等待延误时间成本组合优化模型，运用模拟退火算法对搬运机作业排序优化模型进行求解，从而提高堆场搬运机作业效率，同时提高顾客服务水平。

本文的主要成果有：

1. 本文研究多目标车辆路线安排问题(MVRP)。由于 MVRP 问题不同目标之间相互制约，使其建模和求解具有一定的难度。文中将带有时间窗的 VRP 中的时间变量引入目标函数，建模时重点考虑运输成本和时间两个目标，满足总成本最小和时间准时两个条件，将时间约束作为目标转化成成本函数，实现了将多目标问题转化成单目标问题的求解方法。

2. 从计算方法上，本文将模拟退火算法应用于该模型的求解，以均衡算法的计算时间和最终解的质量，并且与先到先服务的排队原则进行了实验比较。

3. 根据搬运机作业排序优化模型的实际情况，改进了确定初始控制参数的方式；通过反复试验，把理论分析与经验法则结合起来，构建了一个科学有效的、适合搬运机作业排序优化模型计算的冷却进度表，显著地缩减了算法进程的执行时间，同时又不影响最终解的质量。

4. 将模型求解与计算机应用结合起来，编制了 C 语言程序，以实现算法的求解过程，并得到了令人满意的试验结果。

由于路径优化问题和模拟退火算法都属于比较前沿的研究领域，其知识体系在不断发展，本文尚需进一步研究解决的问题有：

1. 集装箱堆场搬运机作业排序优化模型过于直观和理想化，尚需进一步完善。

2. 模拟退火算法是一种随机近似算法，所以求解的质量有赖于大量的试验。随着问题规模的增大和对解的质量要求的提高，所需时间也随之增长，而冷却进度表并不能完全提高算法的效率，解决的办法是实现算法的并行操作以提高其性能。目前常用的并行策略有操作并行、试验并行、区域分裂并行和混乱松弛并行等四种，在实际应用中，可以根据具体问题选择不同的并行策略以进一步提高算法的效率。

3. 模拟退火算法的控制参数对算法性能有一定的影响，至今还没有一个适合于各种问题的参数选择方法，只能依赖于问题进行确定。对于这些参数的选择还需要进一步的研究。

4. 模拟退火算法的应用非常广泛，但是它的理论还不够完善，这也阻碍了它的发展。理论研究的突破必将进一步推进算法本身的发展。

参考文献

1. 张晓萍, 颜永年等. 现代生产物流及仿真. 北京: 清华大学出版社, 1998. 1
2. 中国交通运输协会. 国外物流发展状况及趋势. 商品储运与养护, 2002(4)
3. 陈戌源主编. 集装箱码头业务管理. 大连: 大连海事大学出版社 1998
4. 赵清江. 一类非遍历复杂车辆调度问题的研究. 浙江大学硕士论文, 2003 年 3 月
5. 程宇, 集装箱码头中数字化堆场系统的若干关键技术研究, 武汉理工大学硕士论文, 2004 年 5 月
6. B. De Bacher, V. Furnon, P. Shaw, P. Kilby and P. Prosser. Solving Vehicle Routing Problems Using Constraint Programming and Metaheuristics[J]. Journal of Heuristics, 2000, (6): 501-523.
7. 黄岚, 王康平, 周春光, 原媛, 庞巍. 基于蚂蚁算法的混合方法求解旅行商问题[J]. 吉林大学学报(理学版), 2002, 40(4): 369-373.
8. 郭耀煌, 李军. 车辆优化调度. 成都: 成都科技大学出版社, 1994.
9. 郭耀煌. 安排城市卡车行车路线的一种新算法. 系统工程学报, 1989, 4(2): 70-78
10. 汪寿阳, 赵秋红, 夏国平. 集成物流管理系统中定位——运输路线安排问题的研究. 管理科学学报. 2000, 3(2): 69-75
11. 祝崇隽, 刘民, 吴澄. 供应链中车辆路径问题的研究进展及前景. 计算机集成制造系统 CIMS. 2001, 7(11): 1-6
12. 袁庆达, 杜文, 周再玲. 带软时间窗的混合车队车辆路线问题的模型和算法研究. 西南交通大学学报. 2000, 36(4): 401-406
13. 谢秉磊, 郭耀煌, 郭强. 动态车辆路径问题: 现状与展望. 系统工程理论方法应用. 2002, 11(2): 116-120
14. Chen X., Wan W., Xu X. Modeling rolling batch planning as vehicle routing problem with time windows. Computers & Operations Research. 1998, 25(12): 1127-1136
15. Chong-Jun Z., Min L., Cheng W., Xiao-Bing W.. Two kinds of 2-OPT algorithm for VRP with fuzzy demand. Tien Tzu Hsueh Pao/Acta Electronica Sinica. 2001, 29(8): 1035 — 1037
16. 姜大立, 杨西龙, 杜文. 车辆路径问题的遗传算法研究. 系统工程理论与实践. 1999, 19(6): 40-45
17. 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. 系统工程理论与实践. 1999, 19(8): 65-69
18. 李军. 有时间窗的车辆路线安排问题的启发式算法. 系统工程. 1996, 114(5): 45-50
19. 李军. 车辆调度问题的分派启发式算法. 系统工程理论与实践 1999, (1): 27-33
20. 李军. 货运车辆优化调度理论与方法研究. 西南交通大学博士学位论文. 2000
21. 李军, 郭强. 车辆路径问题的改进表上作业法. 西南交通大学学报. 2000, 35(5): 532-534
22. 李军, 郭强, 刘建新. 组合运输的优化调度. 系统工程理论与实践. 2001, (2):

117-121

23. 李军, 谢秉磊, 郭耀煌. 非满载车辆调度问题的遗传算法. 系统工程理论方法应用. 2000, 9(3)
24. 谢秉磊, 李军, 郭耀煌. 有时间窗的车辆调度问题的遗传算法. 系统工程学报. 2000, 15(3): 290-294
25. 袁庆达, 杜文, 周再玲. 带软时间窗的混合车队车辆路线问题的模型和算法研究. 西南交通大学学报. 2000, 36(4): 401-406
26. 张涛, 王梦光. 遗传算法和 3-opt 结合求解带能力约束的 VRP. 东北大学学报. 1999, 20(3): 253-256
27. 李军, 郭耀煌. 物流配送: 车辆优化调度理论与方法. 北京: 中国物资出版社, 2001
28. 张红历. 基于地理信息系统的车辆优化调度研究及软件实现. 西南交通大学硕士学位论文. 2001
29. Laporte G., Mercure H., Nobert Y.. An exact algorithm for the asymmetrical capacitated vehicle routing problem. Networks. 1986, 16:33-46
30. Eilon S., Watson-Gandy C.D.T., Christofides N.. Distribution management: Mathematical modelling and practical analysis. London: Griffin, 1971
31. Christofides N., Mingozzi A., Toth P.. State space relaxation procedures for the computation of bounds to routing problems. Networks. 1981, 11
32. Desrosiers J., Soumis F., Desrochers N.. Routing with time windows by column generation. Networks. 1984, 14
33. Laporte G., Nobert Y., Desrochers M.. Optimal routing under capacity and distance restrictions. Operations Research. 1985, 33
34. Martello S., Toth P.. Generalized assignment problem. in: Martello S., Toth R(eds), Knapsack problems algorithm and computer implementations. Chichester: Wiley. 1990
35. Clarke G., Wright J.W.. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research. 1964, 12
36. Solomon M.M.. Algorithms for the vehicle routing and scheduling problem with time windows constraints. Operations Research. 1987, 35
37. Lin S.. Computer solutions of the travelling salesman problem. Bell system Technical Journal. 1998, 44
38. Stewart W.R., Golden B.L.. Stochastic vehicle routing: A comprehensive approach. European Journal of Operational Research. 1983, 14
39. Gendreau M., Laporte q Seguin R.. An exact algorithm for the vehicle routing problem with stochastic demands and customers. Transportation Science. 1995, 29(2)
40. 钟小鹏. 动态车辆路径问题实时策略与技术支撑分析. 西南交通大学硕士论文. 2003 年 6 月
41. 谢金星, 邢文训. 网络优化. 北京: 清华大学出版社, 2000.
42. 陈国良. 遗传算法及其应用. 人民邮电出版社, 1999
43. 周培德. 寻求中国货郎担问题最短回路的多项式时间算法. 北京理工大学学报, Vol. 20, No. 2
44. 马良. 旅行推销员问题的算法综述. 数学的实践与认

- 识, Vol.30.No.2.April.2000, 156—165.
45. 金海和, 陈剑, 唐政. 基于 Hopfield 网络学习的多城市旅行商问题的解法. 系统工程理论与实践, 2003 No.7 100-105.
46. 张纪会, 高齐圣, 徐心和. 自适应蚁群算法. 控制理论与应用, Vol.17, No.1, Feb., 2000
47. 王凌, 智能优化算法及其应用, 清华大学出版社, 2001
48. 李静, 列车编组计划及其模拟退火算法, 华北工学院计算机应用专业, 2001
49. 康立山, 谢云等, 非数值并行算法(第一册), 科学出版社, 1994

致 谢

本文是在导师孙小明副教授的悉心指导下完成的。两年多来，孙老师渊博的学识、严谨求实的治学态度和睿智宽容又不乏幽默的学者风范无时无刻不影响着我，并且将永远是我前进道路中的楷模。本文凝聚了导师的大量心血，在论文选题、确定框架到写作、修改的全过程，得到导师悉心的指导，导师的意见和建议给予了我良多启发和帮助。在此向导师孙小明副教授致以衷心的感谢和诚挚的祝愿。

另外，还要感谢项目组成员张东汉、郑柯威、李志、王剑许和杨正娥硕士对本人的关心和帮助，能够与你们共同经历这样一段快乐而充实的日子，对我来说弥足珍贵，我将永远不会忘记，并衷心祝愿你们前程似锦。

特别要衷心感谢我的父母亲和我的妻子陈继芳，在我攻读硕士学位的两年多时间里，他们从精神上和经济上全力支持我，使我能安心钻研自己的学业！

最后，对研究生期间的所有授课老师，以及关心、帮助过我的老师、同学、朋友和亲人表示衷心的感谢。

刘义苍

二零零五年二月
于上海交通大学

攻读学位期间发表的学术论文

- [1] 优化算法在集装箱堆场吊车装卸作业排序中的应用, 工业工程与管理, 2004 年 11 月 (已录用), 刘义苍 孙小明