



**RESEARCH ON THE VEHICLE ROUTING PROBLEM WITH
SIMULTANEOUS PICK-UP AND DELIVERY
WITH TIME WINDOWS**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF BUSINESS ADMINISTRATION
INTERNATIONAL CHINESE COLLEGE**

**GRADUATE SCHOOL, RANGSIT UNIVERSITY
ACADEMIC YEAR 2019**



考虑带时间窗和送取货的
车辆路径规划研究



此论文为申请中国国际学院
工商管理专业研究生学历
之学术论文

兰实大学研究生院
公历 2019 学年

Thesis entitled

**RESEARCH ON THE VEHICLE ROUTING PROBLEM WITH
SIMULTANEOUS PICK-UP AND DELIVERY WITH
TIME WINDOWS**

by

BO WANG

was submitted in partial fulfillment of the requirements
for the degree of Master of Business Administration

Rangsit University
Academic Year 2019

Asst. Prof. Chen Ao, Ph.D.
Examination Committee Chairperson

Assoc. Prof. Yang Shu Chen, Ph.D.
Member

Assoc. Prof. Zheng Jiang Bo, Ph.D.
Member and Advisor

Approved by Graduate School

(Asst.Prof.Plт.Off. Vannee Sooksatra, D.Eng.)
Dean of Graduate School
June 18, 2020

致谢

时间飞逝，两年的泰国学习生活即将画上句号。在兰实大学我认识到很多的同学与老师，从他们身上学到很多，我受益匪浅。

在此我要诚挚的感谢我的论文指导老师郑江波教授，在论文重新选题方面给了我明确的方向，在论文撰写期间不厌其烦的回答我提出的问题，一遍遍的讲解，帮助我解决问题。在论文期间郑老师组织我们一起进行了讨论会，教会我们在讨论中发现新论文中新的亮点与问题。郑老师对学术研究严谨的态度与工作上一丝不苟的精神，值得学习。

在此我还要对我的好朋友赵同学致以真诚的感谢，在过年期间帮助我进行代码的编写，并在后期耐心的与我沟通将代码进行修改，在他的帮助下我的论文顺利完成计算机仿真运算过程。

最后要感谢所有的任课老师的辛苦付出，感谢评审老师对论文题注的指正，感谢中国国际学院的所有老师在学习与生活方面的帮助。感谢所有关心过和帮助过我的老师和同学们，衷心祝愿你们一切顺利，身体健康。

王博
研究生

6105638 : Bo Wang
 Thesis Title : Research on the Vehicle Routing Problem with
 Simultaneous Pick-up and Delivery with Time Windows
 Program : Master of Business Administration
 Thesis Advisor : Assoc. Prof. Zheng Jiang Bo, Ph.D.

Abstract

With a rapid development of the current social economy, the requirements for logistics have become more stringent. The most common problem in logistics activities is vehicle route planning, also known as VRP. How to make a reasonable tip plan for a vehicle with limited resources and complete the transportation plan with minimal costs is one of the resolutions to VRP problems. At present, manufacturers strictly require suppliers' logistics services, such as the time window and other requirements. After studying the operation mode of Company A, it was found that the vehicle route planning requires suppliers' time windows and simultaneous delivery. Therefore, this article will use the Company A's vehicle routing planning (VRPSPDTW) model with time windows and simultaneous delivery as an example to establish a mathematical model. The genetic algorithm was used to calculate the vehicle transportation plan of Company A. By comparing the scheme obtained by the algorithm operation with the original scheme of Company A, an increase in vehicle loading rate increase and a decrease in the number of departures were found, which proved the feasibility and the availability of the mathematical model and algorithm. This research proposed companies to apply similar operating models for practical solutions, and adopts genetic algorithm to optimize the vehicle routing problem with simultaneous pick-up and delivery with time windows.

(Total 73 pages)

Keywords: Time Windows, VRP, VRPSPDTW, Genetic Algorithm

Student's Signature Thesis Advisor's Signature

6105638 : 王博
论文题目 : 考虑带时间窗和送取货的车辆路径规划研究
专业 : 工商管理硕士
指导老师 : 郑江波博士

摘要

随着现在社会经济的飞速发展，对物流的要求也愈加严格。在物流活动中最常见的就是车辆路径规划问题，车辆路径规划问题又称为VRP问题，如何在有限的资源下，对车辆进行合理的出行计划，以最小的运输成本完成运输计划是VRP问题的目标之一。目前制造业对供应商的物流服务要求愈加严格，比如时间窗等方面的要求。本文是在了解A公司的运营模式后，发现其车辆路径规划既有时间窗要求，又有在供应商处同时进行送取货的要求，因此本文将A公司带有时间窗和送取货的车辆路径规划（VRPSPDTW）模式为例建立数学模型利用遗传算法对A公司的车辆运输方案进行运算。通过将算法运行得出的方案与A公司原有方案进行对比，可以发现车辆装载率提高，发车次数减少，证明了数学模型与算法的可行性与有效性。为运营模式相似的企业提供实际工作中的优化方案,即采用遗传算法优化带有时间窗与送取货的车辆路径问题，同时为VRPSPDTW问题的解决提供了实际可操作的方法。

（共73页）

关键词：时间窗；VRP；VRPSPDTW；遗传算法

学生签字.....指导老师签字.....

目录

	页
致谢	i
英文摘要	ii
中文摘要	iii
目录	iv
表目录	vi
图目录	vii
第 1 章	
绪论	1
1.1 研究背景	1
1.2 研究意义	3
1.3 主要研究内容与方法	4
1.4 论文主要创新点	6
第 2 章	
相关文献综述	7
2.1 车辆路径问题	7
2.2 常见算法分类	17
2.3 遗传算法	20
第 3 章	
A 公司现状以及问题描述	23
3.1 A 公司车辆运输配送方案的现状描述	23
3.2 公司车辆运输配送方案的问题描述与分析	25
3.3 车辆运输配送路径问题的改善思路	26
3.4 本章小结	28

目录 (续)

	页
第 4 章	29
车辆路径规划问题的数学模型建立	
4.1 基本假设	29
4.2 集合、决策变量与参数设置	30
4.3 函数构造	32
4.4 本章小结	37
第 5 章	38
A 公司遗传算法结果分析	
5.1 遗传算法计算步骤	38
5.2 相关数据整理与分析	40
5.3 实验结果分析	41
5.4 本章小结	44
第 6 章	45
总结及展望	
6.1 主要结论	45
6.2 研究展望	46
参考文献	48
附录	51
附录	
遗传算法编写程序	51
个人简历	73

表目录

表		页
2.1	VRPSPDTW 要素分类表	16
3.1	车辆路径问题的优化	28
5.1	节点信息表	40
5.2	遗传算法参数解释	41
5.3	参数设置	42
5.4	最优路线表	42
5.5	优化数据对比	43



图目录

图		页
2.1	旅行商问题示意图	8
2.2	多旅行商问题示意图	8
2.3	硬时间窗函数图像	10
2.4	软时间窗函数图像	11
2.5	混合时间窗函数图像	12
2.6	算法分类图	18
5.1	遗传算法步骤图	39



第 1 章

绪论

1.1 研究背景

现在物流行业的发展推动着全球经济的蓬勃发展，也成为推动我国国民经济的持续发展、产业与企业升级重组的重要推动力。在当今社会下物流已经渗透在我们日常的生活，物流承担着不可或缺的责任。2019 年前三个季度我国社会物流总额达到了 215.9 万亿元，全社会物流总费用达到了 10.2 万亿元，同比增长 7.5%；运输费用 5.4 万亿元，同比增长 7.0%；保管费用 3.5 万亿元，同比增长 8.2%；管理费用 1.3 万亿元，同比增长 7.8%。我国全社会物流总费用占 GDP 的 14.7%，相比于美国、日本等发达国家社会物流费用仅占 GDP 的 8%。并且我国若在物流成本方面降低 1%~2%即可产生社会经济效益 1000 到 2000 亿元。这表明我国的社会物流成本还存在很大的下降空间，提高物流效率会带来可观的经济效益。从以上数据我们可以得知前三个季度中运输费用占社会物流总费用 52.9%。目前我国的运输结构中，公路运输货运量多年来占我国货运总量的 50%以上，因此对运输车辆做好合理的路径规划与调度，对于降低公路运输费用有着重要的现实意义。

物流的是指货物从上游供应商流向下游接收处的实际物体的流动过程，在物品的流动过程中产生了装卸、搬运、储存等一系列活动，这些活动产生的费用就构成了物流成本，而物流成本支出又是企业运营成本的组成部分，因此物流成本的高低直接影响了企业的盈利能力强弱。为了提高企业的盈利能力，越

来越多的人将目光聚焦在如何合理利用车辆资源、制定车辆路径规划等问题以降低物流成本，提高盈利能力。在学术界对车辆路径问题（Vehicle Routing Problem, VRP）的研究从未停止过脚步。目前有越来越多的学者从不同的角度去对车辆路径问题进行分析，比如运筹学、计算机科学等不同的角度。

在现实生活中车辆路径问题与我们的生活息息相关，比如城市物流、送餐路线等问题都可以看做是 VRP 问题或者 VRP 问题的延伸与扩展。这些问题都可以通过对车辆的运输与配送进行规划，达到提高运输效率与降低物流成本的效果。VRP 问题最早是由西方学者 Dantzing 和 Ramser（1959）提出，在问题提出的初期考虑的影响因素相对单一，已经无法满足当下社会的复杂需求。随着现在社会信息化程度的提高，与准时化生产（Just in Time, JIT）要求越来越高，因此带时间窗和同时送取货的车辆路径问题（Vehicle Routing Problem with Simultaneous Pick-Up and Deliver with Time Windows, VRPSPDTW）渐渐步入人们的视野之中。相较于传统的 VRP 问题与单一约束下的带时间窗的车辆路径问题（Vehicle Routing Problem with Time Windows, VRPTW）问题和同时送取货的车辆路径问题（Vehicle Routing Problem with Simultaneous Pick-Up and Deliver, VRPSPD），VRPSPDTW 将时间窗与同时送取货两个约束条件放在一起，使得其在进行实际求解时过程时的复杂程度也就大大加深。

本文将主要以 A 公司作为研究对象，在以实体企业为研究对象的基础上对 VRPSPDTW 问题进行深入的了解与分析。A 公司是新大洲本田摩托有限公司在太仓市建立的生产基地，该生产基地是 2018 年投入使用的本田在全球最先进的生产基地之一。A 公司为了满足日常生产的需要，在生产基地附近发展了许多供应商，这些供应商负责给生产基地提供零部件或半成品。根据正常的生产计划安排车辆运输，车辆从生产基地不远的车场出发，携带沿线供应商所需要的货物量，到供应商处卸下相应的货物，并将供应商准备好的货物装车运送到生

产基地。随着准时化生产（JIT）的快速发展，A 公司对供应商的要求越来越严格，除了最基础的提供货物外，还对供应商提供的货物进行全面的质量管控与严格的车辆服务时间控制。A 公司要求货物必须在规定的时间内送达至供应商处，若在指定时间之前送达则会付出少量的成本代价；若在指定时间之后送达将会面临高额的罚款。A 公司现有的车辆路径计划可以满足其需求，但却存在发车频次高、装载率低、路线重复等系列问题，造成了资源的浪费，使得运输成本居高不下。生产企业对于时间要求的严格，也就导致更多人将目光聚集在如何在客户规定的时间内完成服务，同时充分利用现有资源，提高效率，降低运输成本方面。这也是目前 A 公司急需解决的问题。

1.2 研究意义

通过对现有文献的阅读与了解，发现现有的关于车辆路径问题的探讨大多仅考虑了单一约束条件，比如仅考虑同时送取货的 VRPSPD 问题，仅考虑时间窗的 VRPTW 问题，仅考虑随机需求的车辆路径问题等。与单一条件相比同时考虑两个约束条件的 VRPSPDTW 问题的研究相对较少。本文研究对象 A 公司的运营模式有其独特性，即在车辆路径规划方面要考虑的因素较为复杂，不但要考虑硬时间窗的约束，还要考虑在供应商处同时进行送取货的服务，其特点就组成了同时带有时间窗和送取货的车辆路径问题。

本文的理论意义在于同时考虑时间窗与送取货的约束条件，从 VRPSPDTW 的角度构建数学模型、设计遗传算法，并将 A 公司的实际数据进行仿真运算，从而对车辆路径进行优化，并丰富 VRPSPDTW 相关理论的应用研究。

本文主要目标是希望为企业简化运输模式，减少发车次数，提高车辆装载率，降低企业运输成本，提高企业盈利能力。为 A 公司以及运营模式相同的企

业提供考虑时间窗和送取货的车辆路径规划问题的可行性优化方案，为 VRPSPDTW 问题提供实际可操作的计算方法。

1.3 主要研究内容与方法

1.3.1 主要研究内容

第 1 章：绪论。提出了本文的研究背景与研究意义。结合我国物流行业的现状，指出合理的车辆路径规划对于降低企业运输成本的重要性，针对 A 公司的实际情况提出本文的优化目标。

第 2 章：相关文献综述。首先本章将车辆路径规划问题的相关文献进行整理，介绍了车辆路径规划问题的提出和相关问题的扩展；其次对常见算法进行介绍与分类，并结合 A 公司的实际情况说明算则遗传算法的原因，并对遗传算法的相关理论的应用进行整理与归纳。

第 3 章：A 公司现状及问题描述。通过对 A 公司的运营模式的了解，总结 A 公司的运营特点与目前在车辆路径规划方面存在的问题，并对存在问题进行分析，针对目前存在的问题提出合理的、可行的解决方案，为下文构建数学模型奠定基础。

第 4 章：车辆路径问题数学模型的建立。结合 A 公司的实际情况提出车辆路径规划的合理的假设以简化模型，并对模型中涉及到的相关参数、集合进行设置，建立约束条件，明确模型优化目标，构建完整的数学模型。为下文的仿真运算做好准备。

第 5 章：算法结果分析。本章对遗传算法的步骤做了介绍，并将 A 公司的

实际运输数据代入算法进行计算，并对算法得出的结果进行分析。最终通过将原有方案与计算得到的方案进行对比分析，验证本文建立的数学模型与选取的遗传算法的可行性与有效性。

第 6 章：总结与展望。对本文的研究内容与要点进行总结，指出本文研究时的不足之处，与未来可以更加深入研究的方向。

1.3.2 主要研究方法

本文主要采用以下三种研究方法：

1) 文献阅读法

笔者通过大量的相关文献的阅读，对本文涉及到的相关理论知识进行梳理。对带有时间窗和送取货车辆路径规划问题研究现状与相关算法应用有一定的了解。在阅读大量文献的基础上，通过针对带时间窗和送取货的车辆路径问题与求解方法的归纳总结，形成文献综述部分。

2) 数学建模法

在对 A 公司的实际运营情况进行了解分析后，首先提出其现有车辆路径方案中存在的问题，并针对问题提出相应的解决对策，通过数学语言，设置相关参数、集合，提出相关的约束条件，确定优化目标，构建出符合 A 公司实际情况的带时间窗和送取货的车辆路径问题的（VRPSPDTW）数学模型。

3) 计算机编程算法

本文将利用 MATLAB R2019b 为算法工具。利用计算机编程，将构建的数

学模型利用遗传算法在 MATLAB 中进行体现，并将 A 公司的实际的运输数据代入计算。通过将运算结果与现有实际方案中的数据进行对比分析。

1.4 论文主要创新点

本文的创新点从理论与实际来讲有两点创新：

首先本文的以 A 公司的实际运营特点为基础进行数学模型的构建。通过对 A 公司的了解，对其运营现状与问题进行分析，针对现有问题提出相应的对策，并对其运营的相关数据进行整理与分析。建立符合 A 公司实际运营情况的数学模型。在模型中将优化目标设定为在满足顾客满意度的前提下，达到运输总成本最小的目标。本文建立的数学模型能够将提高客户满意度与降低企业车辆运输成本之间的矛盾。

其次本文在基于对 A 公司实际数据进行运算，并将运算结果与实际运输数据的对比分析，对使用遗传算法解决 VRPSPDTW 这类 NP-hard 问题的有效性进行验证。

第 2 章

相关文献综述

2.1 车辆路径问题 VRP

车辆路径规划问题（Vehicle Routing Problem, VRP）最早于 1959 年由 Dantzing 和 Ramser 两人提出，主要研究内容为配送中心需要向需求不同的多家客户提供送货服务，送货服务由配送中心的车辆负责，希望通过对车辆路线的合理安排，使得在货物全部送达的基础上，并在其他条件的约束下，达到车辆行驶总路程最短、用时最少、运输费用最小等目的。在现实生产运作中，大多数运输部门会采用合理的车辆路径规划来提高车辆使用率、降低运输成本，来达到提高企业盈利能力的目的。

1) 旅行商问题 (Travelling Saleman Problem, TSP)

旅行商问题也可以称为货郎担问题，即 TSP 问题，是运筹学中著名的问题之一。将旅行商问题具体化，可以表述为有一个旅行商（在 TSP 问题中对旅行商的自身能力不做限制）从某个城市出发，需要到达已知的所有城市进行交易，并且每个城市仅经过一次，最终旅行商回到出发的城市，问应该如何选择行驶路线使得总路程最短。TSP 的最终目标是对旅行商的行驶路线进行优化，使得其行驶的总路程最短。

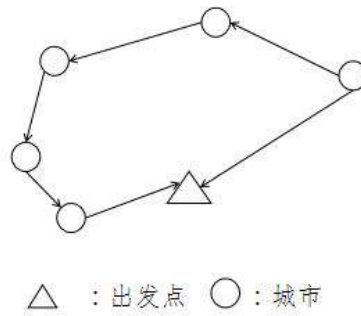


图 2.1 旅行商问题示意图

资料来源：根据钱颂迪《运筹学》整理

2) 多旅行商问题 (Multiple Travelling Saleman Problem, m-TSP)

多旅行商问题是在题在 TSP 的基础上，将单一旅行商增加到多个旅行商，并且城市数量也有所增加，所有旅行商都要在统一起点出发，每一个城市有且仅有一个旅行商经过，在旅行商遍历所有城市后所有旅行商需要返回最初的起点。m-TSP 与 tsp 问题一样，对旅行商自身的能力不走限制，显然与实际情况不相符合，因此 m-TSP 在实际应用中大多是作为对 VRP 问题研究的过渡问题。

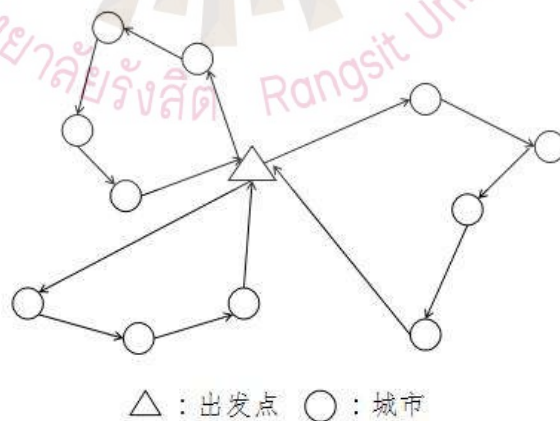


图 2.2 多旅行商问题示意图

资料来源：根据钱颂迪《运筹学》整理

2.1.1 带时间窗的车辆路径问题 (VRPTW)

近年来随着准时化生产模式的普及，客户对于时间方面的要求也愈加严格，因此带有时间窗的车辆路径问题也就应运而生，软时间窗的概念由 Campbell 和 Savelsbergh (2006) 提出并指出带有时间窗的车辆路径问题为 NP-hard 问题。带有时间窗的车辆路径问题即在传统的车辆路径问题的基础上对服务的时间加以限制，更贴合现实的运作情况。时间窗即供应商与客户约定好的服务时间，供应商需要在约定好的时间内为客户提供服务，该时间段可以用 $[ET_i, LT_i]$ 表示，其中 ET_i 表示约定最早提供服务的时间， LT_i 表示约定的最晚提供服务的时间，倘若未能在规定的时间内完成相应的服务，供应商将会面临相应的惩罚。

时间窗可以根据服务时间与具体惩罚方式分为三种类型：硬时间窗、软时间窗与混合时间窗。

1) 硬时间窗 (Hard Time Windows, HTW)

硬时间窗要求供应商必须在于客户规定好的时间内提供服务，只要实际提供服务的时间不在时间窗内，供应商都会面临相应惩罚。硬时间窗对时间要求极其严格，缺乏弹性。一旦供应商在时间窗之外提供服务，客户有权根据合同提出巨额赔偿或拒收服务。对于目前采用准时化生产的企业而言，供应商提前将货物送达回产生不必要的库存费用、管理费用等；供应商延迟服务时间可能会造成企业生产线的延误与闲置。对于企业而言提前提供服务带来的损失要远远小于因延迟服务而带来的损失。

图 2.3 即为硬时间窗下的惩罚函数 (Penalty Function) 图，一旦供应商在与客户约定的时间窗之外提供服务，则会面临惩罚函数值 $p(t)$ ，惩罚函数值为常

数 M ，供应商可以通过对常数 M 的大小用来判断客户对服务时间的重视程度。

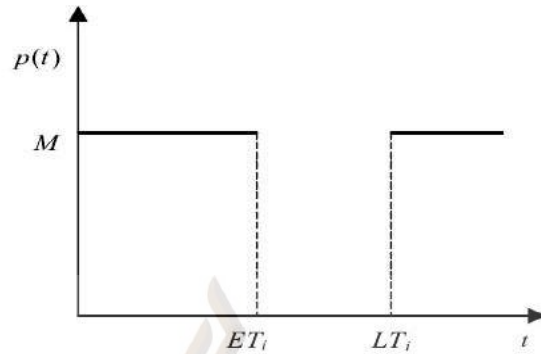


图 2.3 硬时间窗函数图像

资料来源：根据王科峰，2012 整理

2) 软时间窗 (Soft Time Windows, STW)

Bodin 和 Laurence (1994) 首次在研究文献中提出带软时间窗的车辆路径问题，在此之前有关于时间窗的研究大多都是硬时间窗的研究。车辆路径问题中的软时间窗主要是指，供应商在规定好的时间窗内为顾客提供服务，则不会惩罚，倘若在规定好的时间窗外提供服务，供应商将会面临相应的惩罚，具体惩罚将会由实际提供服务时间于惩罚因子决定。在软时间窗的约束下，惩罚系数与违约时间不一定成比例关系。在软时间的约束条件下，企业对供应商的包容度更大，供应商会得到企业服务好坏的评价，而在硬时间的约束下只存在能否完成服务的评价。

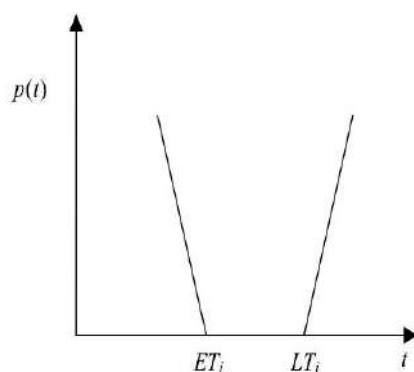


图 2.4 软时间窗函数图像

资料来源：根据王科峰，2012 整理

3) 混合时间窗 (Mixed Time Windows, MTW)

目前很多企业仅采用一种时间窗进行管理，大部分都会选择采用混合时间窗。采用混合时间窗，在规定的窗口 $[ET_i, LT_i]$ 之间进行服务惩罚系数为 0，但在规定的窗口外还存在另一个时间段 $[EET_i, LLT_i]$ ($EET_i < ET_i < LT_i < LLT_i$)，如图 2.5 所示若服务时间 $[EET_i, ET_i]$ 和 $[LT_i, LLT_i]$ 之间，则惩罚系数会根据具体服务时间而定，而服务时间在 $[EET_i, LLT_i]$ 之外，则惩罚系数为任意大的常数 M 。混合时间窗存在某一时间段内客户满意度会根据服务时间的变化而变化。相比硬时间窗来说灵活性更大。

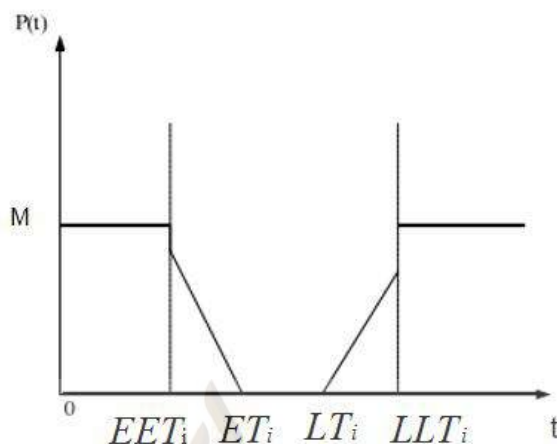


图 2.5 混合时间窗函数图像

资料来源：根据王科峰，2012 整理

王莉、李大卫和王梦（1998）在研究 VRPTW 问题时在前人的研究基础上对遗传算法的编码方式进行了改进，设计出新的交叉算子。最终的结果既考虑到距离问题，也考虑到时间窗的问题。证明使用交叉算子对解决一些组合优化问题的有效性。也为遗传算法去解决组合优化问题提供了新的思考方向。

范志强和乐美龙（2011）在面对带软时间窗多式联运的问题时，采用随机规划的数学方法，建立了带有软时间窗的多式联运路径的数学模型，并利用计算机设计了遗传算法进行求解，最终通过测试，证明其算法的有效性。

2.1.2 同时送取货的车辆路径问题（VRPSPD）

VRPSPD（Vehicle Routing Problem with Simultaneous Pick-up and Delivery）是在 VRP 的基础上在送取货方面进行的延伸。同时送取货的车辆路径问题主要内容是：配送中心的车辆不仅要将中心仓库的货物送至客户处，而且要在客户处进行取货送回配送中心的行为，并且两者之间不可进行独立服务。

Min（1989）首次提出 VRPSPD 的概念，Min 在在文献中解决的具体问题是

图书馆书籍配送，图书馆的两辆车要对 22 个提房图书馆进行书籍的配送与回收，Min 在文献中是将所有的地方图书馆分为了两个相互独立的部分，每个部分由一辆车负责，并将每部分转化成旅行商问题进行求解。

Zachariadis、Tarantilis 和 Kiranoudis (2009) 在对 VRPSPD 问题进行求解时采用将禁忌搜索 (TS) 和指导性本地搜索法 (GLS) 相结合的方法，扩大了搜索范围。在实际求解过程中，首先利用节约算法得到初始解，然后再利用改进的算法对初始解进行优化改进。文章中证明该方法可以对 50-400 个客户点进行优化。

张涛、余绰娅、刘岚、邵志芳和张玥杰 (2011) 在对同时送取货的随机旅行时间车辆的路径进行优化时，以节约算法算出的结果为基础，再次利用分散搜索法对接货进行优化，在仿真实验中证明该算法的有效性。

黄秋爱和李珍萍 (2014) 在对装卸一体的车辆问题 (SDP) 进行求解时采用改进的遗传算法对其进行求解，即采用插入禁忌算法对其进行求解，其最终求解目标为车辆总的行驶距离最短，运算结果证明其算法可实现其设定的目标，并相应的缩短的实际计算时间。

目前企业面临的车辆路径问题大多数与 Min 提出的问题模式相似，车辆在供应商处面临同时送取货的问题。VRPSPD 和 VRPTW 问题都是由基础的 VRP 问题上，根据现实情况增加约束条件产生的新的车辆路径问题，在实际运用中有着极高的现实意义。

在传统的物流中仅考虑车辆的单向运输，在返程的过程中会经常会出现车辆空载的问题，造成极大的资源浪费。通过对车辆行驶路线的合理规划，将送取货结合在一起。可以有效的减少车辆空载现象的发生，提高车辆装载率，降低运营成本，提高企业的盈利能力。

VRPSPD 的最终的目的是在满足客户服务要求的基础上,通过对车辆合理的路径规划,达到发车次数尽可能少,行驶总距离尽可能短的目的。从客户对送取货要求的特点可以将 VRPSPD 分为四种情况:先送货后取货、混合送取货、同时送取货、服务点之间的送取货。

先送货后取货的情况中,首先送货点与取货点并不在同一处,其次车辆是在满足所有送货点的情况下再去取货点取货,直至达到车辆装载上线后回到起点;混合送取货的情况中,送货点与取货点也是不在同一处,区别在于送取货顺序方面,在混合送取货方面对送取货的顺序不做约束,但要确保车辆在行驶过程中不会出现超载现象,最终车辆在满足所有客户的需求后回到起点;同时送取货的情况中,取货点与送货点在同一处,车辆在到达客户处采取先送后取的顺序,在满足所有客户的需求后返回出发点;服务点之间的送取货,此时进行送取货的操作者为第三方物流,即供应商将配送业务外包的情况,第三方不进行任何货物发出与收取的操作,仅在各个服务点之间进行送取货服务。

在车辆路径问题的发展过程中,VRPSPD 问题发展时间相对较短,并且其复杂程度对比经典的车辆路径问题与带时间窗的车辆路径问题也复杂的多,因此目前对 VRPSPD 问题的相关研究也相对较少。

2.1.3 带时间窗的同时送取货车辆路径问题 (VRPSPDTW)

VRPSPDTW 问题是将 VRPSPD 问题与 VRPTW 问题结合在一起,形成的新的 VRP 问题的延伸问题。VRPSPDTW 问题同时较传统的 VRP 问题而言,同时具有时间窗与送取货的约束条件,相比单一约束条件下的 VRP 问题,其复杂程度无疑大大提升。VRPSPD 问题被认为是 NP-hard 问题,无可否认的是加上时间窗限制的 VRPSPDTW 问题也是 NP-hard 问题。NP-hard 问题一直是学术界的热门话题,一般情况下,这类问题的规模较大,约束条件复杂,计算工程量大,

甚至得不到数学意义上的最优解，只能在对参数进行设置的条件下才能得到趋于最优解的满意解。

李玲、谷寒雨和陈坚（2003）在 PDPTW 问题基础上讨论了复杂 PDPTW 的应用求解，其研究规模更大并且受到多车库、多货物类型的影响。并首次采用插入启发式算法对此类问题进行求解，并证明插入启发式算法对解决复杂 PDPTW 问题的有效性。

贾永基等人（2003）提到了对带时间窗的装卸货问题 PDPTW（Pick-up and Delivery with Time Windows, PDPTW）的定义：每一个服务对象都与一个卸货点、一个装货点以及货品的种类和规格，卸货点与装货点的位置可以不同，但车辆必须在规定时间窗内完成卸货与装货的动作。本文默认对于一个供应商而言，其卸货点与装货点是一样的，毕竟相对于不同供应商之间的距离来讲同一供应商装卸货点的距离近的多。

吴璟莉（2006）在求解复杂又时间窗装卸货问题时将其定义成车辆从车场出发，要在规定的时点完成对所有访问点的访问，最终车辆返回车库，其中访问点由供应商的送货点与取货点组成。本文研究与其研究的最大不同点在于，客户的送货点与取货点在同一处，毕竟对于不同客户的地理位置而言，同一个客户的送货点与取货点的距离可以忽略不计。

孙小年、陈幼林和杨东援（2007）为解决物流园区中实际车辆路径安排过程中，存在车辆要完成送货任务的同时又要完成取货的任务的情况，因此将此类问题归纳为 Vehicle Routing Problem with Backhaul（VRPB），即装卸一体化车辆路径问题。

Chopra 和 Meindl（2007）将运输网络分为四类:直送网络、越库网络、专用网路和循环取货网络。Milk Run 模式即循环取货在最初是指英国的农场牛奶配

送模式：送奶工人在农场取走准备好的牛奶并根据各家的需求进行配送，同时取走各家的空瓶；现在指一辆货车按照已路线和约束好的时间窗，依次到供应商处卸下货物并将供应商处的货物装车，并将所有货物运回车场或仓库中的一种公路运输方式。

张倩和丁根宏（2012）将 PDPTW 问题定义为整数线性规划问题，并采用混合分组编码智能算法，并在算法中采用遗传算法粒子群算法想结合的方法。算法结果也证明了该算法相较于改进型贪心算法和改进遗传算法的优越性。

目前对于 VRPSSPDTW 研究的文献相对较少，但现有文献中，不同学者对于该问题的切入点有所不同，本文将根据其要素对其进行分类，结果如表 2.1 所示：

表 2.1 VRPSSPDTW 要素分类表

VRPSSPDTW 构成要素	种类一	种类二
客户需求量	送取货需求量相同	送取货需求量不同
需求已知性	送取货需求已知	送取货需求不确定
需求一致性	同时有送取货的需求	只有送或取货的需求
地理位置	送取货地理位置相同	送取货位置不同
优化目标量	单目标	多目标
货物种类	送货物、取货物	送货物、取包装或送包装、取货物
车型	单车型	多车型
车场数量	单车场	多车场
行驶路线	开放路线	闭合路线
对应关系	汽车与顾客一对多	汽车与客户多对多

资料来源：根据徐俊，2018 整理

本文接下来主要是针对 A 公司的车辆配送路径进行优化，因此根据其现有运输特点，将 VRPSPDTW 定义为：运输部门根据采购计划制定车辆行驶路径，运输部门拥有车型相同的车辆若干，车辆从车场出发，在规定的时间内到达每一个服务点进行送取货服务最终返回到生产基地，每个服务点都同时具有送货与取货的需求，且送取货需求量不一定相同。每个服务点有且仅有一辆车为其提供服务，但一辆车可以服务多个服务点。车辆在行驶的全部过程中不允许出现超载现象。

本文的主要研究目标是在合理利用现有资源的情况下，通过合理的车辆路径规划，既能提高客户满意度，又能降低企业的运输成本，提高企业盈利能力。

2.2 常见算法分类

目前人们使用的算法繁多，根据算法的具体精确程度进行分类，现有算法大致可以分为两类：精确算法与启发式算法。下图是对常见算法进行的归纳与总结，以便能清晰梳理算法之间的关系，如图 2.6 所示：

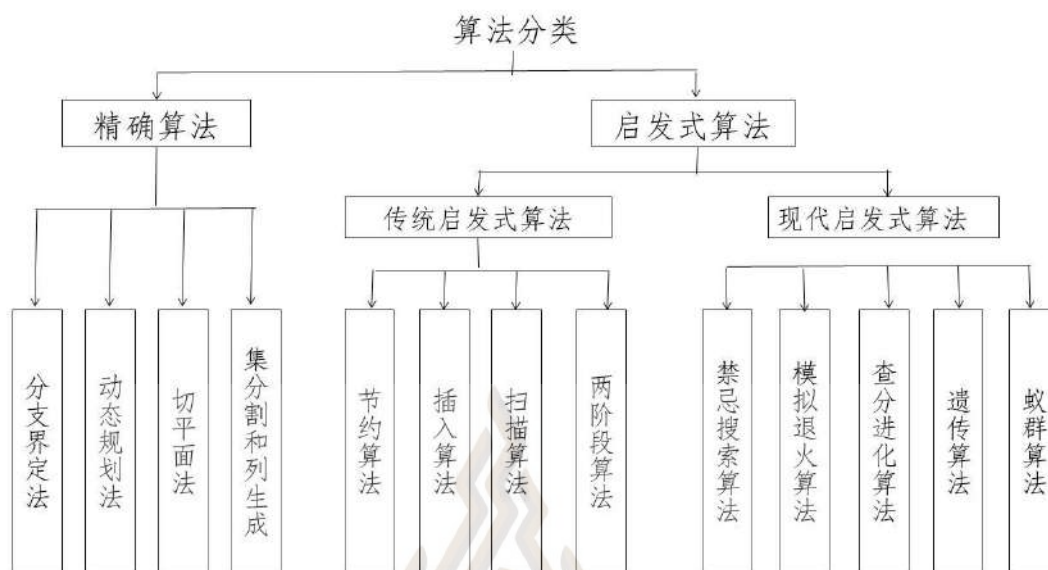


图 2.6 算法分类图

资料来源：姚怡，2016

由图 2.6 可以知道分支定界法、动态规划法、集分割和列生成是精确算法的主要组成部分。精确算法大部分采用线性规划与整数规划等方法来表示数量之间的关系，并以此为基础得到问题的最优解。方金城和张岐山（2006）曾证明该算法在搜索效率方面表现良好，但计算的时间会受到问题变量的影响，变量越多计算时间越长，因此精确算法一般用来解决规模较小的问题，在面对规模较大、约束过多的问题，其实用性不高。

由于现代科技的不断发展，启发式算法通过计算机技术得到了较快的发展。启发式算法可以分为两大类，一类是传统启发式算法，一类是现代启发式算法。相较于精确算法而言，启发式算法的优点在于，在面对复杂问题时其运算时间大大缩短；缺点在于运用启发式算法的结果，有可能是在认为的参数设定下得到的满意解，而不是实际意义上的最优解。追求效率的现代社会，时间成本同样重要。

传统的启发式算法主要是基于运筹学中的数学逻辑以及量化分析原理发展

起来的，传统启发式算法主要由节约法、插入法等算法构成。是较为成熟的计算方法。Clarke 和 Wright (1964) 在文献中提出了成本节约算法。运用此类算法可以得到满意解，但在遇到数据规模较大的问题时，求解时间亦会增加。

现代启发式算法也被称为智能优化算法，主要是以计算机为运算工具，包括禁忌搜索法、模拟退火算法、遗传算法等等。这些算法主要是通过生物界学、物理学等领域得到启发，并将其根据特点进行改进从而运用到现实问题中。现代启式算法在解决复杂的现实问题、计算规模大等问题方面有着良好的表现。张伟 (2015) 利用现代启发式算法中的蚁群算法对车辆路径问题进行优化，利用蚁群算法是正反馈并行机制，对车辆路径问题进行求解，证明其在解决车辆路径问题时求解速度快、并行能力强等优势。

在利用精确算法对问题进行求解时可以得到问题的最优结果，但当问题规模较大的时候回耗费较长的运算时间。VRP 问题是需要快速得到结果的追求时间成本最小化得到问题，因此精确算法在解决 VRP 问题方面并不适用，因此现在人们选择现代启发式算法，利用计算机对问题进行求解，尽管大部分情况只能得到在人为约束下的“最优解”，但与精确算法所耗费的时间成本来说，启发式算法的经济效益更高。

VRPSPDTW 问题是将 VRPTW 问题与 VRSPD 问题柔和在一起的复杂性问题。VRSPD 问题求解的过程较为复杂，Dethloff (2001) 证明 VRSPD 问题是 NP 问题，而 VRPTW 也被证实为 NP 问题，所以将两者进行融合的 VRPSPDTW 问题无疑是 NP-hard 问题,这类问题的复杂程度高，计算规模大，约束条件多，若采用精确算法花费的时间过长，而采用启发式算法会节省大量的时间成本。

本文的研究对象 A 公司在车辆路径方面存在的问题可以概括为 VRPSPDTW 问题，因此下文会针对 A 公司的实际情况建立模型，并选择垫带启发式算法中

的遗传算法进行计算机仿真计算，通过对计算过程中参数的设定，求解在一定参数范围限制下的最优解。为解决 VRPSPDTW 问题提供新的解决思路。

2.3 遗传算法

遗传算法（Genetic Algorithm, GA）是现代启发式算法的重要算法之一，其主要是根据自然界的生物选择和进化机制演变而来的算法（王向，2007）。GA 的主要思路是：先根据问题制定好基因编码的规则，其次对使用其他方法得到的初始解进行处理，按照生物学中优胜劣汰、适者生存的原则，让染色体片段不断的通过交叉、变异产生向最优解趋向的解，并通过判断选择在约束条件下的最优方案，利用其解决实际问题。

GA 的中心思想是通过自然界的遗传现象，使得下一代的子代种群比上一代的父代种群更加优秀，即新的可行解更加接近最优解，最优解为遗传进化中出现的最佳个体或种群。遗传算法凭借简单通用的特性在优化方案等领域有着极佳的表现。

在现有的研究中，大部学者在采用遗传算法进行计算时都采用了对其进行改进使得算法更加贴合实际比如彭春林、梁春华和周泓（2008）、罗勇和陈治亚（2012）、张毅、代恩灿和罗元（2016）等人在解决车辆路径问题时，根据面对的实际情况对遗传算法进行相应的改进，使得算法更加适用于解决问题；廖良才、王栋和周峰（2008）和罗庆和周军（2016）在解决问题时采用了混合的遗传算法，使得在搜索范围更广，等得到的结果更加准确。以上作者可以对遗传算法进行改进和混合使用算法，是因为遗传算法特点，根据葛继科、邱玉辉、吴春明和蒲国林（2008）的文章将遗传算法特点进行简单的总结：

- 1) 处理范围广：由于遗传算法无法直接对参数进行处理，因此需要对参数

进行语言的转化即编码处理，从而得到个体基因。GA 对要处理的问题的数学要求不高，在面对线性、非线性、离散等问题时都可以进行处理。这使得遗传算法在可以处理各种大规模的不同类型的问题。

2) 可操作性强：GA 操作起来简易性强，在 MATLAB 等工具中的应用中也具有智能性，对于参数较多的非结构化问题处理比其他算法更有优势。直接对参数编码的集合进行处理，而不是对问题本身进行处理。

3) 鲁棒性：鲁棒性是来自英文 Robust，是健壮性与稳健性的意思。主要是指在利用计算机对问题进行求解时，不论问题的规模有多大，软件在对其进行处理是都能保持稳定性。

4) 自习性与自适应性：再用遗传算法对进行问题求解时，在个体编码、适值函数设定以及交叉算子和变异算子引用后，算法将智能地根据输入的原始数据，进行自我学习和适应各种规则算式，并自行组织多种信息。

5) 并行性、搜索全面：GA 具有并行计算的能力，能同时对多组初始种群进行处理，这体现了遗传算法比其他方法更优的全局搜索特性。在计算方面，大大节约了运算时间。

6) 可扩展性：GA 的包容性强，可以根据实际的问题对遗传算法进行改进，得到可行性更强的计算方法。

Potvin 和 Bengio (1996) 在研究 VRPTW 问题时，根据自然生物界中的染色体交叉遗传变异的规则，提出了遗传路径系统的概念，通过计算机科学有效的解决了问题并与其他启发式算法进行对比，证明其可操作性与有效性。

王兆赓、李建更和程世东 (2006) 在对集送一体化的车辆路径进行优化是

时采用了两点交叉、多点随机变异等方法来选择遗传操作。大大的缩短了计算时间，并且其在解决多车型的车辆路径方面也有较好的表现。

张萌和曲建华（2015）优化易腐品冷链配送网络的建立包括配送中心选址与配送路径选择两个要素，对遗传算法进行逆转化操作与采用精英策略的改进，求解出配送中心的位置与最佳的配送路线。

陈成（2018）为了解决传统遗传算法计算路径时出现早熟收敛问题，对算法中的交叉变异进行改进，以此来解决带时间窗的车辆路径问题。对比传统的算法，改进后的算法能逃出局部收敛获得最优解。

范厚明、刘文琪、徐振林和耿静（2018）在求解多中心车辆路径问题时，针对传统编解码方式引起的染色体长度不固定导致计算效率低下和易产生不可行解与难以平衡不同进化时期种群中精英比例与种群多样性间、搜索深度与搜索广度间的关系等问题，采用了混合式的遗传算法将编发方式与自适应搜索范围进行改进，通过实验例证和对比分析，验证了算法的有效性。研究成果解多中心联合配送车辆路径问题提供一种新思路。

第 3 章

A 公司现状以及问题描述

3.1 A 公司车辆运输配送方案的现状描述

3.1.1 A 公司的基本运营现状

本文的研究对象 A 公司是新大洲本田在设立在江苏太仓市于 2018 年投入使用的生产基地。新大洲本田是 2001 年成立的中日合资企业，成立至今因其优质的产品质量与良好的信誉一直受到广大消费者的追捧。A 公司在 2019 累计销量为 82.14 万辆，比 2018 年的 73.99 万辆的销量同比增长 10.01%。A 公司的近五年年销售量在中国市场一直名列前茅。

本文主要分析新大洲本田在太仓市建立的生产基地 A 公司，该基地主要负责将供应商提供的半成品或零件进行整车组装，并将成品运至各大销售网点。为了更好的为生产基地提供服务，A 公司在附近设立了车场，负责零部件与车半成品的运输保障。目前车场中为了满足多家供应商的运输需求，拥有许多车型一致的车辆。本文将车辆路径优化的目光聚焦在江苏省内的供应商。

3.1.2 运营模式分析

通过与 A 公司人员的沟通，对 A 公司的运营现状有了大致的了解，现将 A 公司的运营特点进行归纳：

1) 准时化生产需求模式

A 公司的生产模式为准时化生产。准时化生产即在正确的时间内生产出需要数量的产品，以此减少企业因生产过剩造成的人力、设备等方面的资源浪费，最大程度降低企业的生产成本。由于是市场拉动的生产模式，A 公司为了及时相应市场的需求与减少不必要的库存积压，采用的是小批量、多频次的采购模式。准时化生产模式中对产生次品的原因和影响产品质量因素进行严格的管控，从而达到全面质量管理。

2) 时间窗的约束

由于 A 公司生产模式的需要，因此供应商选择上面有着及其严格的要求。首先对供应商的产品质量有着严格的验收标准；其次对货物送达时间有严格要求，即必须在规定的时间内将货物送至供应商处，且完成相应的送取货任务。车辆若未能按时完成任务可能会面临巨额的罚款甚至是拒绝服务。严格的时间窗限制要求车辆要尽可能在规定时间内完成服务，甚至付出少量成本代价提前送达货物，减少因未能按时完成服务而付出大量罚金的现象。

3) 循环取货 (Milk Run)

A 公司目前的配送模式是车辆从车场出发时并携带沿线供应商所需要的货物，在到达供应商处后首先需要将周转箱卸下，然后将供应商准备好的零部件或半成品运输至生产基地。车辆在供应商处先进行卸货然后进行取货活动，与现有运输模式中的循环取货模式类似。

4) 运输需求已知

采购计划是安排车辆运输计划的基础。由于 A 公司的采购计划是周期性制定的，所以不同周期的采购计划是独立存在的，也就导致不同周期间的运输计划的独立性。车辆运输计划是根据采购计划进行安排，对于运输部门来说，不

同周期内各个供应商的物资需求量是已知的。

3.2 A 公司车辆运输配送方案的问题描述与分析

1) 运输模式繁琐

在 A 公司现有的运输计划中，车辆与供应商的关系是多对多的，即一个供应商可以由多辆车提供服务，一辆车也可以服务多个供应商。从表面上看多对多的关系可以充分利用车辆提高装载率，但会存在车辆行驶路线重复的现象，无形中增加了车辆的可变成本。A 公司目前的采购模式为小批量、多频次，供应商与车辆多对多的关系会给运输部门进行车辆路径安排增加难度，也增加了车辆在运输安排中的不确定，导致车辆与人力资源等方面的浪费。

2) 时间管理概念模糊

A 公司目前的采购模式是小批量、多频次，生产方面采用的是 JIT 生产模式。生产模式决定了采购模式。由于 JIT 生产模式对生产线上有着及其严格时间的把控，因此对供应商送达货物时间也有这严格的限制。在于供应商签订的合同中就有明确的说明，若 A 公司无法保障在规定时间内提供所需数量的优质产品，则会面临着高额的罚款。虽然两者之间对时间的管理十分严格，在合同中对时间的界定模糊不明确。

3) 装载率偏低

A 公司目前的运输模式为多对多的关系，看上去可以充分利用车辆，提高装载率，但在实际操作中会出现重复送货或空车运输的情况，造成不必要的资源浪费。由于所需产品的材质、质量等对装载要求有所不同，会影响车辆的装载率，并且到供应商处送货与取货的先后顺序也会对整体的车辆装载率有所影

响。目前 A 公司的车辆装载率保持在 70%左右，其装载率还未达到最优情况，还有很大的提升空间。

4) 供应商安排混乱

近几年 A 公司发展迅速，合作的供应商也随之增加，目前与之合作的供应商遍布周围各大省市。随着供应商数量的不增加，随之而来的运输部门的车辆路径安排也就愈加复杂。目前在车辆运输安排方面采取的是统一管理，并未对供应商进行分类管理，会出现部分车辆与人力既要负责省内又要负责省外的混乱局面。目前的车辆安排计划存在管理混乱的风险。

3.3 车辆运输配送路径问题的改善思路

通过上述对 A 公司运营情况的了解与目前其车辆路径规划方面存在的问题进行的分析，接下来将针对以上存在问题提出针对性的改进措施，具体措施如下所示：

1) 简化运输配送模式

目前 A 公司的车辆运输模式中车辆与供应商之间是多对多的关系，为简化车辆运输配送模式，先对车辆与供应商的对应关系进行改进，规定一辆车在不超出承载范围内可以为多家供应商提供服务，但一个供应商有且仅有一个车辆提供服务的模式，且车辆从车场出发在满足沿线供应商所有需求后返回到生产基地。即将车辆与供应商关原有的多对多的模式改进为一对多的模式，这样可以降低车辆的不确定性情况。

2) 设定硬时间窗

A 公司目前的 JIT 生产模型，对时间管理的严格要求，若设定硬时间窗会更加符合实际情况。因此将 A 公司与供应商之间的约定设置为硬时间窗。严格要求车辆在规定时间内完成一系列服务，以确保生产能够连续不间断的进行下去。倘若由于车辆安排的失误未能按时提供服务，则 A 公司会面临巨额罚金。

3) 提高装载率

目前 A 公司的车辆与供应商的关系为多对多，原意是合理利用车辆的空空间，提高装载率，但多对多的关系相对复杂，在进行车辆路径安排时会出现车辆单程空载的问题。本文将会在满足供应商需求的前提下，对车辆运输配送方案进行修改。由于单周期内供应商的需求量是已知的，发车次数的减少，会提高车辆装载率。因此本文的目标之一是通过减少发车次数，提高车辆装载率。

4) 分区管理供应商

由于目前与 A 公司稳定合作的供应商数量较多，为了方便对供应商进行管理与方便运输部门制定车辆配送路径的计划，提高车辆装载率，本文将根据供应商的地理位置进行划分，将供应商按照地理位置进行分区管理。本文将目光放在江苏省内的供应商，并对其进行车辆路径配送的规划。

表 3.1 车辆路径问题的优化

		原有方案	改进措施
运输安排	车辆与供应商关系	多对多	一对多
	运输要求	车场出发最终返回生产基地	
时间安排	服务时间	包括卸箱、装箱、装车等时间	设定硬时间窗

表 3.2 车辆路径问题的优化（续）

供应商	供应商管理	统一管理	按照地理位置进行 分区管理
	运输计划	周期性计划，需求确定	
优化目标	车辆运输成本最小化、发车次数最少、行使距离最小化		

资料来源：笔者自行整理

3.4 本章小结

本章主要通过对 A 公司人员的访谈，对 A 公司现有的运营模式进行总结，并对相关的数据进行整理与分析。通过对运营情况的了解与具体数据的分析对 A 公司目前车辆运输计划存在的问题进行深入的分析与总结，并根据总结出的问题，有针对性的提出相关解决对策，为下文建立数学模型奠定基础。



第 4 章

车辆路径规划问题的数学模型建立

4.1 基本假设

通过上述章节中，对 A 公司现有车辆配送方案的分析，针对其存在的问题提出相应的解决对策，再结合本文简化运输模式，减少发车次数，提高车辆装载率，降低企业运输成本，提高企业盈利能力的目的，为了降低不可控因素对模型的影响突出研究目标，本文提出以下假设：

- 1) 需求量：由于车辆运输计划是根据采购计划制定的周期性计划，所以运输部门在进行运输计划时，所有供应商的需求量是已知的；
- 2) 地理位置：车辆所要经过的所有节点的地理位置是已知的；
- 3) 车型：车场拥有的车辆车型相同；
- 4) 车辆：所有车辆均为自有车辆，不考虑业务外包的影响；
- 5) 服务时间：不同车辆在同一个节点处提供的服务时间相同；
- 6) 运输费率：车辆的单位运输费率不受车辆实际载重量的影响，并且运输费率为常数；
- 7) 送取货顺序：所有车辆应先提供送货服务再进行取货服务，先送后取；
- 8) 对应关系：一辆车可以为多个节点进行服务，而一个节点仅能接受一辆

- 9) 车提供服务，车辆与节点的对应关系为一对多；
- 10) 载重量：在任何时候车辆的实际载重量不超过车辆最高载重量；
- 11) 交通：道路情况良好，不考虑交通堵塞等特殊情况；
- 12) 甩挂：所有车辆均为一车一挂，不存在甩挂的情况；
- 13) 供应商：本文中仅考虑江苏省内供应商的情况；

上述假设是根据影响 A 公司车辆路径问题的因素所提出来的，并对其中不可控因素进行控制以简化数学模型，下文中对数学模型的建立是在上述假设成立的前提下进行构建的。希望可以利用数学模型对 A 公司的车辆路径问题进行优化。

4.2 集合、决策变量与参数设置

4.2.1 集合

本文的数学模型会涉及到大量的集合，集合整理如下所示：

$V_0 = \{1, 2, \dots, n-1, n\}$ ，表示供应商、车场与生产基地的集合。其中 1 表示车场，n 表示生产基地，集合中 2 到 n-1 代表供应商的节点；

$V = \{2, 3, \dots, n-1\}$ ，表示所有供应商的节点，并且 $V \cup \{1, n\} = V_0$ ；

$K = \{1, 2, \dots, m\}$ ，表示全部车辆的集合，数字代表运输计划中车辆按照出发顺序编号，m 即为运输计划中最后一辆车的编号；

v ：表示 V 的子集，有 $v \subseteq V_0$ ，且其的模的长度为 $|v| \leq |V| = n$ 。

4.2.2 决策变量

x_{ijk} : 车辆 $k(k \in K)$ 的行驶变量, 可表示为 $x_{ijk} \in [0,1]$, 若节点 i 到节点 j 之间的运输由车辆 k 负责, 那么 $x_{ijk} = 1$, 否则 $x_{ijk} = 0$, 并且 $i \in V_0; j \in V_0; i \neq j$;

y_{ik} : 车辆 $k(k \in K)$ 的运输配送服务变量, 可表示为 $y_{ik} \in [0,1]$, 如果节点 i 由车辆 k 为其提供服务, 那么 $y_{ik} = 1$, 否则 $y_{ik} = 0$, 其中 $i \in V_0$;

L_{ik} : 表示车辆 $k(k \in K)$ 离开节点 i 时, 车辆的实际载重量, 并且在运输的全部过程中, 车辆的实际载重量不允许超过车辆承载能力上限, 其中 $i \in V_0$;

S_{ik} : 车辆 $k(k \in K)$ 到达节点 i 的时间, $i \in V_0$, 由于所有车辆都从编号为 1 的车场出发, 所以 $S_{1k} = 0$;

4.2.3 参数设置

模型中会使用到大量的参数, 本节会对参数含义进行解释:

Q : 车辆的装载能力, 为了便于模型的理解, 不考虑货物包装的影响, 将车辆装载能力转化为能容纳 1 m^3 的周转箱箱数, 车辆装载上限为 50 个周转箱, 即 $Q = 50$;

c_f : 车辆发车成本, 大于 0 的常数, 单位: 元/辆;

c_v : 车辆的单位运输成本, 大于 0 的常数, 单位: 元/km;

d_{ij} : 节点 i 与节点 j 之间的距离, $i \in V_0; j \in V_0; i \neq j$;

b_i : 车辆在供应商 i 的取货箱数, $i \in V_0$;

d_i : 车辆给供应商 i 送的周转箱数, $i \in V$;

ET_i : 节点 i 规定开始服务时间, $i \in V$;

LT_i : 节点 i 规定最迟的开始服务时间, $i \in V$;

t_i : 车辆在节点 i 处提供服务所耗费的时间, $i \in V_0, t_1 = 0, t_n = 0$,

t_{ij} : 车辆从节点 i 到节点 j 之间车辆的行驶时间, $i \in V_0; j \in V_0; i \neq j$;

α : 衡量车辆固定发车成本与可变成本的参数, $\alpha \in [0,1]$;

M : 一个用于判定的任意大的常数;

4.3 函数构造

根据 A 公司的实际情况, 模型的最终目标是在满足客户的情况下, 使得运输总成本最小化。而在本文中客户满意的判断标准主要体现在车辆能否在规定的时间内完成相关服务; 而运输总成本的最小化主要体现在固定的发车成本与运输成本之和的最小化。

4.3.1 目标函数

由于运输总成本由固定成本与可变成本量部分组成, 可表示为: 总运输成本 $TC = \text{固定成本 } FC + \text{可变成本 } VC$ 。其中:

1) 固定成本 FC: 发车成本

$$\sum_{j \in V} \sum_{k \in K} c_f x_{1jk} \quad (4.1)$$

2) 可变成本 VC: 车辆总的运输距离

$$\sum_{i \in V_0} \sum_{j \in V_0} \sum_{k \in K} c_v d_{ij} x_{ijk} \quad (4.2)$$

3) 总目标函数

$$\min Z = \alpha \sum_{j \in V} \sum_{k \in K} c_f x_{1jk} + (1 - \alpha) \sum_{i \in V_0} \sum_{j \in V_0} \sum_{k \in K} c_v d_{ij} x_{ijk} \quad (4.3)$$

在上述假设中，将单个车辆的固定发车成本假设为 c_f ，发车次数为 x_{1jk} ，总固定成本为 $c_f x_{1jk}$ ，由于车辆的固定发车成本为常数，因此总固定成本主要受发车次数 x_{1jk} 的影响；车辆在节点 i 到节点 j 之间的距离表示为 d_{ij} ，单位运输成本表示为 c_v ，总的可变成本表示为 $c_v d_{ij} x_{ijk}$ ，单位运输成本为大于零的常数，因此总的可变成本主要受到车辆行驶距离 d_{ij} 的影响。

在对 A 公司进行了解时发现，A 公司目前在运输总成本方面发车的固定成本占较大比重，可变成本即运输费率占比相对较小，现有运输费用的占比方式使得企业更偏向于减少发车次数，并且使发出去的车辆尽可能的加长行驶距离。由于 α 系数的赋值会对结果产生加大的影响，本文将根据企业的实际情况将 α 的范围设定在 0.5-0.7 之间。结合本文简化运输模式，减少发车次数，提高车辆装载率，降低企业运输成本的目标，所以最终得到的运输总成本表达式为 (4.3)。

4.3.1 约束条件

本小结主要是根据实际情况将车辆、供应商、时间窗等条件进行约束：

1) 对车辆与供应商之间的约束

$$\sum_{j \in V} x_{1jk} = \sum_{i \in V} x_{ink}, \forall k \in K \quad (4.4)$$

$$\sum_{k \in K} y_{ik} = 1, \forall i \in V \quad (4.5)$$

$$\sum_{i \in V_0} X_{ijk} = y_{jk}, \forall j \in V; k \in K; i \neq j \quad (4.6)$$

$$\sum_{j \in V_0} X_{ijk} = y_{ij}, \forall i \in V; \forall j \in V; k \in K; i \neq j \quad (4.7)$$

约束式 (4.4) 是对车辆的起始点与终点的约束。其表示所有车辆 k 从编号为 1 的车场出发，在满足沿线所有供应商的需求后，车辆需要返回到编号为 n 的生产基地；

约束式 (4.5) 是对供应商与车辆对应关系的约束。其表示一个供应商有且仅有一辆车为其提供服务，一辆车可以为多个供应商节点提供服务，即车辆与供应商之间为一对多的关系；

约束式 (4.6) 和约束式 (4.7) 是对车辆行驶路径的约束。其表示车辆 k 从车场出发后，所有车辆 k 都不会重复经过同一个供应商进行服务。

在车辆与供应商之间的约束，首先通过约束式 (4.4) 与约束式 (4.5) 对车辆进行约束，然后通过约束式 (4.6) 与约束式 (4.7) 对车辆的行驶路径进行了约束。

2) 对车辆装载能力的约束

$$L_{1k} = \sum_{i \in V} b_i y_{ik}, \quad \forall k \in K \quad (4.8)$$

$$L_{jk} = L_{ik} + \sum_{j \in V} x_{ijk}(d_i - b_j) - M(1 - y_{jk}), \quad \forall k \in K, \forall i \in V_0, i \neq j \quad (4.9)$$

$$0 \leq L_{ik} \leq Q, \quad \forall k \in K, \forall i \in V_0 \quad (4.10)$$

约束式 (4.8) 是对车辆从车场出发时携带周转箱箱数的约束。其表示车辆 k 从车场出发时携带的周转箱箱数要与车辆 k 所负责的所有供应商的需求数相同，即车辆携带周转箱箱数要满足沿线供应商的需求量；

约束式 (4.9) 是车辆 k 在供应商 j 处进行送取货服务后的实际装载箱数。实际装载箱数为到达 j 供应商的前一个供应商 i 处携带的箱子数减去在 j 点卸下的箱子数加上在 j 供应商取的的箱子数。当 j 、 k 不符合条件是，任意常数 M 会发挥总用，使得 $L_{jk} < 0$ 。

约束式 (4.10) 是对车辆载重量的约束。其表示车辆的实际载货量是小于等于车辆承载上限，并且车辆的实际承载量也不存在小于 0 的情况。并且其与约束式 (4.9) 相互制约。

在对车辆装的约束上，首先确保车辆装载量能够满足供应商的需求，其次保障了车辆在运输途中不会出现超载或空载的现象。

3) 对硬时间窗的约束

$$S_{jk} = S_{ik} + \sum_{j \in V} x_{ijk}(t_i + t_{ij}) - M(1 - y_{ij}), \quad \forall k \in K; \forall i \in V_0 \quad (4.11)$$

$$ET_i \leq s_{ik} \leq LT_i, \forall k \in K; \forall i \in V_0 \quad (4.12)$$

约束式 (4.11) 表示车辆在对节点 i 服务后到达下一个节点 j 点的时间。到达节点 j 的时间为车辆到达节点 i 点的时间、车辆在节点 i 消耗的服务时间与车辆在供应商 i 与 j 之间车辆行驶时间之和。 M 在公式里的作用与上述约束式中 (4.9) 相同。

约束式 (4.12) 表示的是车辆 k 到达节点 i 点的时间。即车辆需要在规定好的硬时间窗内到达节点 i 处并完成相关服务。

约束式 (4.11) 与约束式 (4.12) 主要在对时间窗上进行了约束，以确保所有车辆都能够在时间窗内完成相应的服务，避免因车辆延误而支付巨额罚金，使得运输方案中的惩罚系数为 0，到达客户满意度最大化的目标，减少不必要的资源浪费。

4) 对其他条件的约束

$$x_{ijk} \in \{0,1\} \quad \forall k \in K; \forall i \in V_0; \quad \forall j \in V_0, i \neq j \quad (4.13)$$

$$y_{ik} \in \{0,1\} \quad \forall k \in K; \forall i \in V_0 \quad (4.14)$$

约束式 (4.13) 表示判断车辆 k 是否负责供应商 i 到供应商 j 之间的线路行驶，若是则 $x_{ijk} = 1$ ，若不是，则 $x_{ijk} = 0$ ；

约束式 (4.14) 表示判断车辆 k 是否负责供应商 i 处的配送任务，若是则 $y_{ik} = 1$ ，若不是，则 $y_{ik} = 0$ ；

4.4 本章小结

本章是基于第 2 章和第 3 章，针对 A 公司的实际运营情况作出相应的假设条件，并对模型中涉及的集合、决策变量与参数等进行设置，并根据优化目标构建目标函数。本章主要针对 A 公司 VRPSPDTW 问题进行数学模型的构建，为第五章运用遗传算法进行运算奠定基础，亦是整篇文章的重要部分。



第 5 章

A 公司遗传算法结果分析

5.1 遗传算法计算步骤

遗传算法是启发式算法中最经常提到的算法之一。相较于精确算法，遗传算法在解决 NP-hard 问题方面有着独特的优势，这类问题一般很难有最优解，但遗传算法可以在一般可行解的基础上通过染色体的交叉、变异，不断向最优解进行计算。若对算法中的参数进行设定，就可以得到参数限制下的最优解。

通过对 A 公司的运营模式的分析，其车辆路径规划问题是一个 NP-hard 问题。因此本文通过对如种群数量、遗传代数等一系列参数进行设定，通过计算机的运行求得在以上参数设定下的最优解。遗传算法主要是采用达尔文生物进化论中适者生存，不适者淘汰的规则，在算法中主要涉及以下六个步骤：

步骤一：由于遗传算法无法直接对数据进行处理，因此要将问题中涉及到的参数进行编码，即将实际问题的结构转变为遗传算法中染色体结构，利用转化后的结构进行运算；

步骤二：将初始种群中产生的一组大小为 N 的初始可行解，作为这一组的初始种群；

步骤三：建立适应度函数，并计算个体适应度；

步骤四：建立适应度函数值，对个体进行筛选，适者生存，不适者淘汰；

步骤五：按照生物学中的遗传规律，对现有种群进行交叉算子与变异算子

的运算，根据下一代的种群当中的个体适应度来进行“优胜劣汰，适者生存”的选择，已得到更加适应环境的种群。

步骤六：判断遗传算法是否满足终止条件，终止条件为遗传代数到达设定代数上限，或者现有结果是参数限定下的最优解。若为满足终止条件，则重新回到步骤三；若满足终止条件，则可以输出最优解。

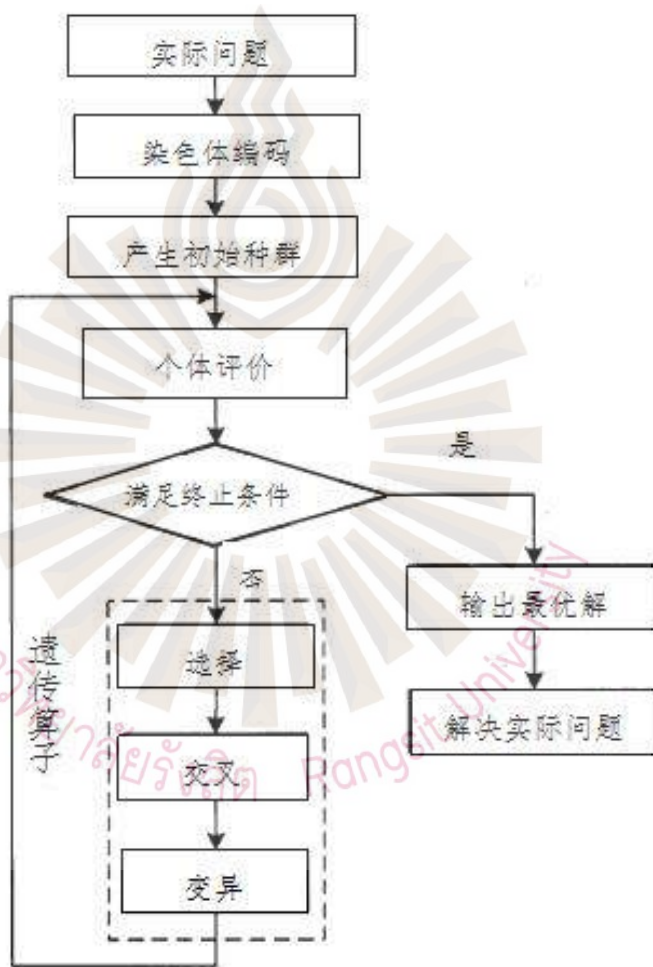


图 5.1 遗传算法步骤图

资料来源：根据葛继科、邱玉辉、吴春明和蒲国林，2008 整理

5.2 相关数据整理与分析

本文在进行运算时采用的是 A 公司某一采购周期内的真实数据。在进行运

算之前，为了确保算法的顺利进行对相关数据进行优化。比如周期内供应商的需求量并非每天都相等，因此对需求量进行了平均计算的方法。本文的重点放在靠近太仓生产基地附近的供应商。

由于在运算过程中会需要大量的供应商的数据，比如供应商的地理位置、供应商送取货的数量、服务时间与时间窗等数据。本文通过对数据的整理，将具体数据呈现在表 5.1:

表 5.1 节点信息表

编号	纬度	经度	ET	LT	t_i	d_i	b_i
1	31.4934	121.0966					
2	31.2826	120.8502	8:30	10:00	20	23	19
3	34.8847	116.8700	8:30	10:30	13	6	5
4	32.6250	119.7783	9:00	10:30	14	20	21
5	34.5986	119.1204	9:00	14:30	25	18	30
6	31.4262	120.5348	9:00	11:00	40	21	13
7	31.3101	120.5817	10:30	11:00	40	44	30
8	31.5201	120.0660	10:30	11:30	28	13	15
9	32.2924	119.9102	10:30	11:30	40	33	43
10	31.9790	120.1873	8:30	10:30	29	25	10
11	31.9492	119.9497	8:30	10:30	40	9	35
12	31.8765	120.1991	8:30	11:00	30	25	18
13	32.7864	119.9882	8:30	11:00	35	15	22
14	31.6526	119.7788	14:00	15:30	50	34	47
15	32.0736	120.0181	14:00	14:30	23	27	9
16	32.0698	119.8619	14:00	15:00	40	12	17
17	32.7118	119.8881	14:30	15:00	30	15	20
18	31.4529	121.1255					

资料来源：笔者自行整理

注：表格中 ET 代表的是车辆最早到达时间； LT 代表车辆最晚到达时间； t_i 代表车辆在 i 处进行服务所需要的时间； d_i 为供应商 i 的需求量； b_i 为车辆取货周转箱数。

5.3 实验结果分析

本文遗传算法的实现是依托于计算机运算完成的，所有的操作在 CPU 为 Intel (R) Core (TM) i5-4210U 2.40Ghz、内存为 4.00GB 的 64 位 Windows10 的操作系统商完成，并在算法工具上选择了 MATLAB R2019b。

从文章中对遗传算法的简介中，了解到在运用遗传算法的过程中会涉及大量的参数。其中种群规模 (Size_Of_Group)、迭代次数 (G)、交叉概率与变异概率是所有遗传算法都会涉及到的基本参数。在对遗传算法实际运行过程中，以上参数的设定会对实际运算范围与运算时间产生直接的影响。在数据较为庞大时，要尽可能的减少运算时间，此时参数的设定就至关重要。在参数的取值范围内控制变量，可以看出不同参数对最终结果的影响，在进行参数选择时要尽可能选择适应度函数最大化与运行时间最短的参数。若存在两组参数，适应度函数的值相似，则选择实际运行短的参数。接下来将会对设计到的参数进行解释说明，具体如表 5.2 所示：

表 5.2 遗传算法参数解释

参数名称	含义
染色体编码	将现实问题映射到编码空间，一个染色体编码表示问题的一个解
初始种群 个体	通过其他方式得到的结果，并将其作为有遗传算法的初始解 种群内每个染色体都是一个个体
迭代次数	算法在达到认为设定的遗传代数就停止运算，防止陷入到无限的运算中
选择算子	交叉算子 父代染色体通过交换部分染色体形成新的子代染色体 变异算子 父代染色体通过改变自身的染色体
个体适应度值	通过适应度函数目标函数值转化为固定值域内的适应度函数， 基于适应度值计算个体被选择的概率

资料来源：根据张晓彤，2018 整理

由于本文选取 A 公司相应数据规模不大，因此根据实际情况对相关参数进行设置。具体如表 5.3 所示：

表 5.3 参数设置

参数	取值
种群规模	100
迭代次数	100
交叉概率	0.75
变异概率	0.1

资料来源：笔者自行整理

根据表 5.3 对遗传算法参数的设定，接下来将 A 公司的实际数据代入要算法中进行计算。得到的最优方案如表 5.4 所示：

表 5.4 最优路线表

车辆	途径供应商	行驶距离 (km)	出发装载率	返回装 载率
1	7	60	88%	60%
2	4-13	54	70%	86%
3	2-12	41	96%	74%
4	14	56	68%	94%
5	9	86	66%	86%
6	15-16	69	78%	52%
7	10-11	54	68%	90%
8	6-8-17	58	96%	96%
9	3-5	48	48%	70%
总行驶距离		526		

资料来源：MATLAB 运行结果

通过最终得到的最优解的方案，可以看出车辆标号为 1、4 和 5 的车辆仅为

一个供应商提供服务，原因有两个：一是由于该供应商的需求量较大，车辆在满足该供应商之后无法再去为其他供应商进行服务；二是由于时间窗的限制，在车辆满足上一个供应商之后，无法寻找到能在时间窗内完成服务的车辆。

除了上述车辆编号为 1、4 和 5 的车辆外，其余车辆均为两个即两个以上的供应商提供服务，在时间窗的范围内，一辆车为多个供应商提供服务，可以充分利用车辆的使用空间并有效减少发车次数。A 公司原有车辆运输方案中发车 12 次，利用算法减少发车次数 3 次。

从车辆行驶的总距离来看，算法得到的最优解方案中车辆行驶总距离为 526km，较 A 公司原有方案中车辆行驶总路程下降 8 个百分点左右。

从车辆的装载率方面来看，部分车辆的返回装载率与车发装载率之间的差距较大，这是由于供应商在送取货数量绵绵数量的不对等高造成的，有些供应商的取货量小，送货需求量大，比如编号为 10 的供应商；有些则相反送货需求量小，取货量大，比如编号为 11 的供应商。理论上讲，通过合理的车辆配送路径进行规划，可以提高车辆的装载率，但由于约束条件的增加，会出现车辆自身还有能力，但由于无法满足时间窗的条件无法为其他供应商提供服务的情况。这也是 VRPSPDTW 问题与单一约束条件下 VRP 问题的区别，受到自身约束条件的限制，无法最大程度的利用车辆装载空间，最大程度的减少发车次数。

表 5.5 优化数据对比

	发车次数	总行驶距离	装载率
原有方案	12	568	69.75%
遗传算法最优解	9	526	77%
优化比率	25%	8%	7.25%

资料来源：笔者自行整理

通过上文对遗传算法得到的最优解与 A 公司现有的车辆配送路径的对比分析，可以得知在利用遗传算法进行优化后 A 公司的车辆运输在总行驶距离、与发车次数方面均有所减少，在车辆装载率方面也有所提升，A 公司可以根据本论文中提出的算法在对车辆配送路径进行优化。在满足客户满意度的前提下，降低企业的运输总成本，提高企业的盈利能力。因此可以推断本文在根据 A 公司的现实情况所构建的数学模型的合理性有有效性，在对遗传算法的选择上有一定的科学依据。可以利用上述数学模型与遗传算法，对带有时间窗与同时送取货的车辆路径问题进行优化。

5.4 本章小结

本章主要对 A 公司的现有车辆路径问题的数据进行整理与分析。将整理过的数据代入到遗传算法中，利用 MATLAB 进行相关的仿真运算，并在过程中不断将参数进行调试使得算法更加优化。最后将利用遗传算法得到的结果与 A 公司现有的运输方案进行简单对比分析。验证了本文建立模型与遗传算法的可行性与有效性。

第 6 章

总结及展望

6.1 主要结论

车辆路径规划问题一直是物流配送环节中重要的一环，是提高企业第三利润源的关键。由于现实情况的复杂性就决定了车辆路径规划问题的复杂性，因此成为了许多学者不断研究的热点。车辆路径问题具有很强的现实意义，可以通过优化提高物流运转效率，降低物流成本，提高企业盈利能力。

本文首先通过文献的阅读，对现有车辆路径问题、算法分类与算法应用进行了大量的整理与归纳。对现有的问题类型与优化方法有一定的了解，并在其中发现针对同时带有时间窗和送取货的车辆路径问题的相关理论与优化方法资料相对较少，因此本文以此为切入点，对 VRPSPDTW 问题进行研究。通过对 A 公司的了解得知，A 公司采用的运输模式与 VRPSPDTW 问题相似，因此本文以 A 公司的车辆路径规划为例进行研究。

通过对 A 公司的运营模式的了解，根据 A 公司的实际运输情况对底油时间窗和送取货的车辆路径问题进行定义，即一批车型相同的车辆，根据已知的运输任务，从车场出发，在规定的时间内为每一个服务点进行服务，服务点同时具有送取货的需求并且送取货的需求量不一定相同，车辆在行驶的过程中，其实际载重量不得超过自身的最高装载量，且每个服务点有且仅有一辆车进行服务，即车辆与服务点的关系为一对多。并根据给出的定义进行假设，设置参数，建立与 A 公司实际相符合的数学模型。希望通过模型达到简化运输方式，

并在满足顾客满意度的前提下，减少发车次数，提高装载率，降低运输成本，提高盈利能力的目标。在算法选择方面采用了遗传算法，在计算过程中不断对算法中涉及的参数进行调试，最终确定适合的参数进行计算。

文章在进行 A 公司车辆路径优化时，首先通过硬时间窗的约束，确保所有车辆在规定时间内到达并完成服务，提高顾客满意度；其次将车辆与供应商的关系改为一对多，简化运输模式，降低车辆重复经过相同路线的情况，车辆总运输路程减少；最终通过计算机仿真，将 A 公司的数据带入到构建的模型中，利用遗传算法进行求解，结果证明在发车次数与车辆总运输路程方面都减少，车辆装载率提高，降低了企业的运输成本。

本文在进行数学模型是根据 A 公司的实际情况进行构建的，在进行计算机仿真运算时采用的是 A 公司某采购周期的真实数据，因此确保了进行遗传算法的可行性；在遗传算法过程中对参数不断进行调整，最终将选择的参数运用到遗传算法中，将 A 公司的实际数据输入，结果表明在确保车辆准时到达的前提下，车辆的装载率有所提高，发车次数有所减少，运输总成本有所下降，证明了遗传算法解决在带有时间窗与送取货的车辆路径规划问题方面的有效性。同时也为面临相似同时带有时间窗与送取货的车辆路径问题的企业，提供实际可操作的车辆路径优化计算方法，补充了同时带有时间窗与送取货的车辆路径规划问题的实际应用研究。

6.2 研究展望

本文主要围绕 A 公司的实际运营情况，研究其带有时间窗的同时送取货的问（VRPSPDTW）题，由于自身能力的限制，本文在研究时仅针对问题的基本情况，在实际情况中影响车辆路径规划的因素更多。因此，本文在今后的研究中还可以在以下几个方面进行优化：

1) 数据选取: 本文所引用的数据均来自 A 公司某一采购周期的数据, 仅涵盖生产基地附近的几家供应商的数据, 并对数据进行相应的处理。整体来讲选取的数据规模较小。

2) 模型理想化: 本文在对数学模型进行构建时, 对影响条件进行了大量的假设, 导致模型过于理想化。比如在实际的运输过程中天气情况、交通堵塞虎屋特殊装载要求、车辆不同型号等都会对车辆路径规划产生影响。

3) 算法选择上: 本文采取的是现代启发式算法中的遗传算法, 对于其他算法仅做了简单的介绍。然而有学者表明, 混合式算法即多种启发式算法结合而成的算法对于解决 NP-hard 问题有着显著的效果。在接下来的研究中, 可以将重点放在融合多种启发式算法形成独特的混合算法, 以此来解决实际问题。



参考文献

- Bodin, L., & Laurence, L. (1994) . Commentary—Visualization in Vehicle Routing and Scheduling Problems. *ORSA Journal on Computing*, 6 (3) , 13 (2/3) , 277-285.
- Campbell, A., & Savelsbergh, M. (2006) . Incentive schemes for attended home delivery services. *Transportation Science*, 40 (3) , 327-341.
- Chopra, S., & Meindl, P. (2007) . *Supply chain management: strategy, planning, and operations*. Upper Saddle River, Pearson Prentice Hall.
- Clarke, G., & Wright, J. W. (1964) . Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12 (4) , 568-581.
- Dantzig, G. B., & Ramser, J. H. (1956) . The Truck Dispatching Problem. *Management Science*, 6 (1) , 80-91.
- Dethloff, J. (2001) . Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *Or Spektrum*, 23 (1) , 79-96.
- Min, H. (1989) . The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research, Part A General*, 23 (5) , 377-386.
- Potvin, J. Y., & Bengio, S. (1996) . The vehicle routing problem with time windows part II:genetic search. *Computing*, 8 (2) , 165-172.
- Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2009) . A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service. *Expert systems with applications*, 36 (2p1) , 1070-1081.
- 陈成. (2018) . 基于改进遗传算法的带时间窗的多目标配送路径优化. *信息技术与信息化*, 011(10) , 48-51.
- 戴晓晖, 李敏强, & 寇纪淞. (2000) . 遗传算法理论研究综述. *控制与决策*, 5 (03) , 263-268.
- 范厚明, 刘文琪, 徐振林, & 耿静. (2018) . 混合粒子群算法求解带软时间窗的VRPSPD问题. *计算机工程与应用*, 54 (19) , 221-229.

- 方金城, & 张岐山. (2006). 物流配送车辆路径问题 (VRP) 算法综述. *沈阳工程学院学报(自然科学版)*, 2 (4), 357-360.
- 范志强, & 乐美龙. (2011). 面向随机环境的带软时间窗多式联运路径优化. *工业工程与管理*, 16 (05), 68-72.
- 葛继科, 邱玉辉, 吴春明, & 蒲国林. (2008). 遗传算法研究综述. *计算机应用研究*, 25 (10), 2911-2916.
- 黄秋爱, & 李珍萍. (2014). 带有次序限制的装卸一体化车辆路径问题研究. *三明学院学报*, 31 (06), 19-24.
- 贾永基, 谷寒雨, & 席裕庚. (2004). 求解 PDPTW 问题的一种快速禁忌搜索算法. *控制与决策*, 10 (01), 57-60.
- 廖良才, 王栋, & 周峰. (2008). 基于混合遗传算法的物流配送车辆调度优化问题求解方法. *系统工程*, 26 (08), 27-31.
- 罗庆, & 周军. (2016). 基于混合遗传算法的物流配送路径优化分析. *中央民族大学学报(自然科学版)*, 25 (03), 50-56.
- 罗勇, & 陈治亚. (2012). 基于改进遗传算法的物流配送路径优化. *系统工程*, 30 (08), 118-122.
- 李玲, 谷寒雨, & 陈坚. (2003). 复杂 PDPTW 问题的插入启发式算法. *计算机工程*, 29 (16), 65-66.
- 庞凌. (2019). 物流配送中烟花算法结合遗传算法的异质车队路径优化方法. *计算机测量与控制*, 27 (08), 245-248.
- 彭春林, 梁春华, & 周泓. (2008). 求解同时取货和送货车辆路径问题的改进遗传算法. *系统仿真学报*, 27 (09), 2266-2270.
- 钱颂迪, & 胡运权. (2012). 《运筹学》. 第四版. 清华大学出版社.
- 孙小年, 陈幼林, & 杨东援. (2007). 装卸一体化车辆路径问题的遗传算法研究. *系统工程理论与实践*, 2 (02), 149-152.
- 王科峰. (2012). 带时间窗分车运输同时收发车辆路径问题及其启发式算法. *运筹与管理*, 21 (2), 83-88.
- 王莉, 李大卫, & 王梦. (1998). 带有交货期窗口的调度及算法. *系统工程学报*, 12

(04), 18-26.

王兆赓, 李建更, & 程世东. (2006). 基于遗传算法的集送一体化的车辆路径问题. *计算机工程与应用*, 04 (01), 208-211.

吴璟莉. (2006). 求解复杂有时间窗装卸货问题的遗传算法. *计算机应用*. 26 (06), 1459-1462.

王向. (2007). 遗传算法综述. *河北经贸大学学报 (综合版)*, 7 (07), 154-155.

徐俊. (2018). 带时间窗和送取货的车辆路径规划问题研究 (硕士论文), 暨南大学, 中国.

姚怡. (2016). 二维装箱问题的启发式算法研究 (硕士论文). 华南理工大学, 中国.

张涛, 余绰娅, 刘岚, 邵志芳, & 张玥杰. (2011). 同时送取货的随机旅行时间车辆路径问题方法. *系统工程理论与实践*, 31 (10), 1912-1920.


张萌, & 曲建华 (2015). 基于改进遗传算法的易腐品冷链配送网络优化. *物流工程与管理*, 37 (07), 139-142.

张伟. (2015). 基于蚁群算法的物流配送中车辆路径优化问题研究. *物流科技*, 38 (10), 48-50.

张晓彤. (2018). 基于改进遗传算法面向多周期订单的动态制造单元构建 (硕士论文), 燕山大学硕士, 中国.

张毅, 代恩灿, & 罗元. (2016). 基于改进遗传算法的移动机器人路径规划. *计算机测量与控制*, 24 (01), 313-316.

张倩 & 丁根宏. (2012). 求解 PDPTW 的混合分组编码智能算法. *重庆理工大学学报 (自然科学)*, 26 (11), 70-74.

The image features a large, faint watermark of the Rangsit University logo. The logo consists of a stylized flame or sunburst shape at the top, with a circular base made of radiating lines. The text 'มหาวิทยาลัยรังสิต Rangsit University' is written in a semi-circle below the logo.

附录

遗传算法编写程序

มหาวิทยาลัยรังสิต Rangsit University

遗传算法编写程序主程序

1、主程序

```
clear;
```

```
close all;
```

```
rng('default')
```

```
rng(1)
```

```
%% 载入数据
```

```
option.F=1;
```

```
data.Node=xlsread('data.xlsx');
```

```
data.numNode=length(data.Node(:,1));
```

```
data.numVehicle=data.numNode-2; %最多三辆车
```

```
data.loadMax=50; %装箱上限
```

```
data.alpha=0.5; %目标权重
```

```
data.v=50*1000; %车辆速度 30km/h
```

```
data.cf=30; %发车固定成本
```

```
data.cv=1; %形成成本/m
```

```
figure
```

```
hold on
```

```
position=find(data.Node(:,9)==1);
```

```

plot(data.Node(position,2),data.Node(position,3),'ro','LineWidth',2,...

'MarkerEdgeColor','k',...

'MarkerFaceColor','b',...

'MarkerSize',10)

position=find(data.Node(:,9)==0);

plot(data.Node(position,2),data.Node(position,3),'rh','LineWidth',2,...

'MarkerEdgeColor','k',...

'MarkerFaceColor','r',...

'MarkerSize',10)

legend('车场','供应商')

%% 计算距离

for i=1:data.numNode

    for j=1:data.numNode

        temp1=data.Node(i,2);

        temp2=data.Node(i,3);

        temp3=data.Node(j,2);

        temp4=data.Node(j,3);

        data.D(i,j)=vdist(temp1,temp2,temp3,temp4);

    end

end

end

```



```
data.D(isnan(data.D))=0;

option.D=(data.numNode-2)*2+data.numVehicle;

option.aimFcn=@aimFcn_1;

option.bound=[0*ones(option.D,1),1*ones(option.D,1)];

%% 目标权重

option.weight=[0.5,0.5];

%% 算法参数

option.peiod=50 ; %Maximum Iterations 最大迭代次数

option.genMax=option.peiod;

option.M=500; %Population Size 种群规模

option.popSize=option.M;

option.show_t=1; %是否显示迭代次数

option.show_pc=0; %是否显示迭代收敛图

option.p1=0.85; %Crossover Rate 交叉概率

option.p2=0.2; %Mutation Rate 变异概率

option.creat_x=@creat_x_1;

option.aimFcn=@aimFcn_2; %平方

option.by=@bianyi_1;

option.TS_len=option.M*5; %禁忌列表长度
```

```

option.c1 = 0.5;      %学习因子 1 APSO

option.c2 = 0.5;      %学习因子 2 APSO

option.wmax = 1;      %最大权重 APSO

option.wmin = 1;      %最小权重 APSO

%%

x=cell(option.popSize,1);

fit=zeros(option.popSize,1);

for j=1:option.popSize

    x{j}=option.creat_x(option,data);

    fit(j)=option.aimFcn(x{j},option,data);

end

data.x=x;

data.fit=fit;

%%

[accord_fit_min2,result2]=GA(option,data);

%%

figure

hold on

plot(accord_fit_min2,'LineWidth',2)

```

```

legend('GA')

xlabel('迭代次数')

ylabel('适应度值')

%%

[~,result]=aimFcn_2(result2.best_x,option,data);

%%

figure

hold on

position=find(data.Node(:,9)==1);

plot(data.Node(position,2),data.Node(position,3),'ro','LineWidth',2,...

'MarkerEdgeColor','k',...

'MarkerFaceColor','b',...

'MarkerSize',10)

position=find(data.Node(:,9)==0);

plot(data.Node(position,2),data.Node(position,3),'rh','LineWidth',2,...

'MarkerEdgeColor','k',...

'MarkerFaceColor','r',...

'MarkerSize',10)

for i=1:length(result.recording.path)

```

```

plot(data.Node(result.recording.path {i},2),data.Node(result.recording.path {i},3),'Line
Width',2)

end

legend('车场','供应商')

% clc;

% clear;

% close all;

function [accord_fit_min,result]=TSGA(option,data)

%% 开始计算

creat_x=option.creat_x;

aimFcn=option.aimFcn;

by=option.by;

%option.popSize=option.popSize*2;

%% 遗传算法求解

% % % 初始种群生成

%

% x=cell(option.popSize,1);

% fit=zeros(option.popSize,1);

% for i=1:option.popSize

%   x {i}=creat_x(option,data);

```

```
% fit(i)=aimFcn(x{i},option,data);

% end

x=[data.x];

fit=[data.fit];

%% 建立本地种群，实现简单的协同进化

x_p=x;

fit_p=fit;

accord_fit_mean(1,:)=mean(fit_p);

accord_fit_min(1,:)=min(fit_p);

TS_list=x;

%% 进化开始

for t=1:option.genMax

    if option.show_t

        t

    end

    %% 选择

    max_fit=max(fit);

    min_fit=min(fit);

    if max_fit==min_fit
```



```
grade=0.5*ones(1,option.popSize);

else

p0=option.p1*(1-0.1*t/option.genMax);

grade=p0*(max_fit-fit)/(max_fit-min_fit)+0.01;

end

jishu1=1;

jishu2=1;

position=zeros(option.popSize*2,1);

while jishu1<=option.popSize*2

if grade(jishu2)>=rand

position(jishu1)=jishu2;

jishu1=jishu1+1;

end

jishu2=jishu2+1;

if jishu2>option.popSize

jishu2=1;

end

end

%% 交叉
```

```

temp_x=x;

for i=1:option.popSize

    if rand<0.5 %离散交叉

        if rand<0.5

            temp=randi(length(x {position(i)}(:,1)));

temp_x {i,1}=[x {position(i)}(1:temp,:);x {position(i+option.popSize)}(temp+1:end,:)];

        else

            temp=randi(length(x {position(i)}(1,:)));

temp_x {i,1}=[x {position(i)}(:,1:temp);x {position(i+option.popSize)}(:,temp+1:end)];

        end

    else %算术交叉

        rand_p=rand(size(x {position(i)}));

        temp_x {i,1}=rand_p.*x {position(i)}+(1-
rand_p).*x {position(i+option.p)};

    end

end

%% 变异

for i=1:ceil(option.popSize*option.p2*(1+t/option.genMax))

    temp11=randi(option.popSize);

```

```

temp_x{temp11}=by(temp_x{temp11},option,data);

end

%% 在禁忌列表中的 x 重新生成

for i=1:option.popSize

    for j=1:length(TS_list)

        if sum(sum(temp_x{i}==TS_list{j}))==length(temp_x{i}(:))

            temp_x{i}=creat_x(option,data);

            break;

        end

    end

end

end

%% 更新禁忌列表

TS_list=[TS_list;temp_x];

if length(TS_list)>option.TS_len

    TS_list(1:(length(TS_list)-option.TS_len))=[];

end

temp_fit=zeros(option.popSize,1);

%% 重新计算适应度

parfor i=1:option.popSize

```

```

    [temp_fit(i)]=aimFcn(temp_x {i},option,data);

end

%% 精英选择

x=[x_p;temp_x];

fit=[fit_p;temp_fit];

[~,position]=sort(fit(:,1));

fit=fit(position,:);

x=x(position,:);

x_p=x((1:option.popSize),:);

fit_p=fit(1:option.popSize,:);

accord_fit_mean(t+1,:)=mean(fit_p);

accord_fit_min(t+1,:)=fit_p(1,:);

%result.best_x=x_p{1};

    %[result.best_fit,result.detail,result.best_x]=aimFcn_1(x_p{1},option,data);

end

%%

if option.show_pc

    %% 适应度曲线

```

```
figure

plot(accord_fit_min(:,1),'LineWidth',2);

hold on;

plot(accord_fit_mean(:,1),'LineWidth',2);

title('总目标函数适应度曲线');

legend('最佳适应度','平均适应度');

xlabel('迭代代数');

ylabel('适应度值');

figure

plot(accord_fit_min(:,1),'LineWidth',2);

hold on;

title('总目标函数适应度曲线');

xlabel('迭代代数');

ylabel('适应度值');

end

%%%

result.best_x=x_p{1};

%[result.best_fit,result.detail]=fit_p(1);
```



```
[result.best_fit,result.detail]=aimFcn(result.best_x,option,data);
```

1、成本函数

```
recording.node{jishu1}=[recording.node{jishu1};temp];
```

```
[flag,tempLoad0]=recordingLoad(option,data,path);
```

```
recording.node{jishu1}(:,12:13)=tempLoad0(1:end-1,:);
```

```
CV(jishu1)=sum(recording.node{jishu1}(:,14))*data.cv;
```

```
CV=sum(CV);
```

```
CF=jishu1;
```

```
fit=data.alpha*CF+(1-data.alpha)*CV;
```

```
%%
```

```
if nargout>1
```

```
    result.recording=recording;
```

```
end
```

```
end
```

2、约束条件

```
function [fit,result]=aimFcn_2(x,option,data)
```

```
%global data
```

```

%% 解码

priority=x(1:(data.numNode-2));

[~,S]=sort(priority);

%%

jishu=1;

jishu1=1;

for i=1:length(S)

    noNode=S(i)+1;

    if jishu==1

        path=[1,noNode];

        D=data.D(1,noNode);

        runT=D/data.v;

        windowsT=data.Node(noNode,[4,5])*24;

        serviceT=data.Node(noNode,[6])/60;

        T0=windowsT(1)-runT;

        T=T0+runT+serviceT;

        demand=data.Node(noNode,[7,8]);

        gapT=windowsT(2)-T+serviceT;

        temp=[1,noNode,T0,T-

```

```
serviceT,gapT,windowsT,serviceT,runT,demand,demand*0,D,T];
```

```
    recording.node{jishu1}=temp;
```

```
    jishu=jishu+1;
```

```
else
```

```
    node0=path(end);
```

```
    D=data.D(node0,noNode);
```

```
    runT=D/data.v;
```

```
    windowsT=data.Node(noNode,[4,5])*24;
```

```
    tempT=T+runT;
```

```
    demand=data.Node(noNode,[7,8]);
```

```
    serviceT=data.Node(noNode,[6])/60;
```

```
    % 检查是否满足装载约束
```

```
    tempPath=[path,noNode];
```

```
try
```

```
    [flag,tempLoad0]=recordingLoad(option,data,tempPath);
```

```
catch
```

```
    a=1;
```

```
end
```

```
if flag==0
```

```
    % 检查当前时间窗是否满足
```

```
if windowsT(1)<=tempT && tempT<=windowsT(2)

    flag=0;

else

    if windowsT(1)>tempT

        gapT=windowsT(1)-tempT;

        position=find(recording.node{jishu1}{:,5}<=gapT);

        if isempty(position)

            recording.node{jishu1}{:,3:4}=recording.node{jishu1}{:,3:4}+gapT;

            recording.node{jishu1}{:,5}=recording.node{jishu1}{:,5}-gapT;

            T=T+gapT;

        else

            flag=1;

        end

    else

        flag=1;

    end

end

end

end

if flag==1

    path=[path,data.numNode];

end
```

```

recording.path{jishu1}=path;

D=data.D(noNode,data.numNode);

runT=D/data.v;

T=T+runT;

temp=[path(end-1),data.numNode,T-
runT,T,0,0,0,runT,demand,demand*0,D,T];

recording.node{jishu1}=[recording.node{jishu1};temp];

[flag,tempLoad0]=recordingLoad(option,data,path);

recording.node{jishu1}(:,12:13)=tempLoad0(1:end-1,:);

CV(jishu1)=sum(recording.node{jishu1}(:,14))*data.cv;

jishu1=jishu1+1;

path=[1,noNode];

D=data.D(1,noNode);

runT=D/data.v;

windowsT=data.Node(noNode,[4,5])*24;

serviceT=data.Node(noNode,[6])/60;

T0=windowsT(1)-runT;

T=T0+runT+serviceT;

demand=data.Node(noNode,[7,8]);

gapT=windowsT(2)-T+serviceT;

```



```
temp=[1,noNode,T0,T-
serviceT,gapT,windowsT,serviceT,runT,demand,demand,D,T];
```

```
recording.node{jishu1}=temp;
```

```
else
```

```
jishu=jishu+1;
```

```
path=[path,noNode];
```

```
T=T+runT+serviceT;
```

```
gapT=windowsT(2)-T+serviceT;
```

```
temp=[path(end-1),noNode,T-serviceT-runT,T-
serviceT,gapT,windowsT,serviceT,runT,demand,demand,D,T];
```

% 起点，终点，出发时间，抵达时间，距离时间窗下线的距离，时间窗，服务时间，行驶时间，需求，当前装载，距离，服务完成时间

```
recording.node{jishu1}=[recording.node{jishu1};temp];
```

```
end
```

```
end
```

```
end
```

```
path=[path,data.numNode];
```

```
recording.path{jishu1}=path;
```

```
D=data.D(noNode,data.numNode);
```

```
runT=D/data.v;
```

```
T=T+runT;
```

```

temp=[path(end-1),data.numNode,T-runT,T,0,0,0,0,runT,demand,demand*0,D,T];

recording.node{jishu1}=[recording.node{jishu1};temp];

[flag,tempLoad0]=recordingLoad(option,data,path);

recording.node{jishu1}(:,12:13)=tempLoad0(1:end-1,:);

CV(jishu1)=sum(recording.node{jishu1}(:,14))*data.cv;

CV=sum(CV);

CF=jishu1;

fit=data.alpha*CF+(1-data.alpha)*CV;

%%

if nargout>1
    result.recording=recording;
end
end

```

3、距离矩阵

```
function d = SphereDist2(x,y,R)
```

```
%根据两点的经纬度计算大圆距离(基于 Haversine 公式)
```

```
%x 为 A 点[经度, 纬度], y 为 B 点[经度, 纬度]
```

```
if nargin < 3
```

```

R = 6378.137;

end

x = D2R(x);

y = D2R(y);

h = HaverSin(abs(x(2)-y(2)))+cos(x(2))*cos(y(2))*HaverSin(abs(x(1)-y(1)));

d = 2 * R * asin(sqrt(h));

end

function h = HaverSin(theta)

    h=sin(theta/2)^2;

end

function rad = D2R(theta)

    rad = theta*pi/180;

end

```

适应度函数

%% 重新计算适应度

```

parfor i=1:option.popSize

    [temp_fit(i)]=aimFcn(temp_x{i},option,data);

end

```

```
%% 精英选择

x=[x_p;temp_x];

fit=[fit_p;temp_fit];

[~,position]=sort(fit(:,1));

fit=fit(position,:);

x=x(position,:);

x_p=x((1:option.popSize),:);

fit_p=fit(1:option.popSize,:);

accord_fit_mean(t+1,:)=mean(fit_p);

accord_fit_min(t+1,:)=fit_p(1,:);

%result.best_x=x_p{1};

%[result.best_fit,result.detail,result.best_x]=aimFcn_1(x_p{1},option,data);

end
```

个人简历

姓名	王博
生日	1996年7月22日
出生地	中国 吉林 松原
教育背景	本科：昆明理工大学津桥学院 专业：物流管理，2018 硕士：RANGSIT UNIVERSITY 专业：工商管理，2020
永久地址	中国江苏省扬州市江都区
邮箱号码	bowang0722@163.com

