****

**本科毕业设计(论文)调研报告**

|  |  |
| --- | --- |
| 题 目： | 基于粒子群算法的车辆路径规划 |
| 学生姓名： | 周子夔 |
| 指导教师： | 何小贤 |
| 学 院： | 计算机学院 |
| 专业班级： | 软件工程2004 |

本科生院制

2024年3月

基于粒子群算法的车辆路径规划

摘要

近年来，我国物流运行保持复苏态势，但物流成本一直居高不下，如何降低物流成本，提高客户满意度一直是业界关注的问题，随着实体经济复苏，物流需求快速增长，进一步完善了物流供应服务体系和供应链。研究表明，运输成本占物流成本的很大一部分，其中燃料和路费约占运输成本的 70%，客户满意度也随着时间和成本的提高而降低。

为解决上述问题，本研究采用改进的粒子群算法(PSO)求解，将实际情况抽象为多目标优化汽车路径问题(VRP)，以总成本最低、满意度最高为优化目标。VRP是给定一系列发货点和收货点，如何组织车辆以最小的总成本或最短的总时间完成所有送货任务，并返回到起始点。

由于 VRP 问题是经典的 NP-难题，使用精确求解器很难得到最优的大规模问题解法，因此这项研究采用了改进的粒子群算法。研究将粒子群初始化，通过混沌映射法提高其多样性；莱维飞行算法用于粒子探索和学习能力的提高；历史最优解采用kopt方法进行了优化，使解的品质得以提升。

这项研究通过高德的城市拥堵数据、Homberger 车载数据集和自适应粒子群算法的对比来验证这一结果的有效性。

**关键词：**PSO VRP 客户满意度 多目标优化

Vehicle Routing Problem Based on Particle Swarm Algorithm

ABSTRACT

In recent years, China's logistics operation to maintain the rebound trend, with the recovery of the real economy, logistics demand is growing rapidly, logistics supply service system and supply chain to further improve, but the logistics cost has been high, how to reduce logistics costs and improve customer satisfaction has been the industry's concern. Research shows that transportation costs account for a large part of logistics costs, of which fuel and road costs account for about 70% of the transportation costs, and customer satisfaction also decreases with the increase of time and cost.

In order to solve the above problems, this study abstracts the practical situation into a multi-objective optimization vehicle routing problem (VRP), which is solved by an improved particle swarm algorithm (PSO) with the optimization objectives of lowest total cost and highest satisfaction.The VRP is a problem of how to organize the vehicles to complete all the delivery tasks at the smallest total cost or the shortest total time, given a series of shipping and receiving points, and to return to the starting point.

Since the VRP problem is a classical NP-hard problem and it is difficult to obtain an optimal solution for large-scale problems using an exact solver, an improved particle swarm algorithm is used in this study. In this study, the chaotic mapping method is used to initialize the particle swarm in order to improve the diversity of the particle swarm; the Lévy flight algorithm is used to improve the exploration and learning ability of the particles; and the k-opt method is used to optimize the historical optimal solution in order to improve the quality of the solution.

In this study, we use the urban congestion data from Gaode Sachs, Homberger vehicle dataset to verify the validity of the results and compare them with the adaptive particle swarm algorithm.

Key words:：PSO VRP Customer satisfaction Multi-objective optimization

1. 绪论

1.1 研究背景

在现实应用中，车辆路径规划除正常耗费之外，往往还要考虑实时路况、时间窗约束、时变车速等多种因素。本专题拟构建优化模型，以最低的总配送费用和最高的平均顾客满意度为目标。

车辆路线问题(VRP)最早于 1959 年由丹齐格(Dantzig)和拉姆瑟(Ramser)(Ramser)首先提出，它是指一定数量的客户，各自对货物有不同数量的需求，由配送中心向客户提供货物，由一个车队负责分配货物， 组织适当的行车路线，目标是使客户的需要得到满足。 并能达到路程最短，费用最少，花费时间最少等一定的约束条件。由定义可知，著名的旅行商问题(TSP)是 VRP 问题的一个特例，VRP 也属于 NP 难题，因为 Gaery 已经证明 TSP 问题是 NP 难题。

VRP自1959年提出以来，因为其应用和经济价值很高，一直收到国内外学者的广泛关注。

VRP问题可以描述如下：

存在一种拥有 M 辆车、拥有 Q 车容积、拥有 N 个客户的配送中心 D，每个客户都有它的需求 D。车辆以场站为起点向顾客提供寄送服务，最后再返回场站，要求顾客全部寄送完毕，每位顾客一次寄送完毕，且以全部车线总车距最小为目的，不得违反车辆容量限制。车辆路线的实际问题有配货中心的配货问题，制定公交线路的问题，寄送信件、报刊的问题，航空、铁路的日程安排问题，工业废料的收集问题等等。

车辆路线问题产生了许多基于车辆基本路线问题(VRP)的不同延伸和变化型态，包括时窗限制车辆路线问题、 追求最佳服务时间的车辆路线问题、多车类型的车辆路线问题、考虑收集到的车辆路线问题、 随机需求的车辆路线问题、车辆路线问题等。以及车辆路线问题，以学术研究为基础，以实际应用为基础。

物流(Logistics)是供应链活动的一部分，以最低的成本，通过运输、保管、配送等方式，实现从商品原产地到商品消费地，从原材料、半成品、成品、服务消费以及相关信息的计划、执行和管理的全过程，以满足顾客的需求。物流以仓储为中心，促进生产和市场的齐头并进。物流行业的参与企业根据自身的定位差异，可分为基础物流资源提供企业和物流资源整合企业。其中，基础物流资源企业主要分布在物流行业各业态及基础设施领域，是物流行业运输网络、节点的主要提供者；物流资源整合企业主要分布在信息系统和物流行业的各业态中。

物流行业目前的运输费用较高，其中占据大头的就是运输费用。从费用结构方面来看，运输费用9.55万亿元，占比为53.77%；保管费用5.95万亿元，占比为33.5%；管理费用2.26万亿元，占比为12.73%。进入 21 世纪以来，国内物流业随着经济全球化格局的形成、物流需求规模持续保持较高增长、物流业增加值稳步增长，已经进入快速发展通道。 2013 年国内社会物流总额已达 197.8 万亿元，同比增长 9.3%，国家统计局数据显示， 但物流成本过高已成为制约物流行业持续健康发展的重要因素。中国物流成本居高不下《中国采购发展报告(2014)》统计数据显示，2013 年中国社会物流总费用达 10.2 万亿元人民币，占国内生产总值(GDP)的 18%，而这一比例在美国仅为 8.5%，在发达国家和发展中国家分别为 9%和 6%。在高额的交通成本构成中，燃油成本占 46%，道路桥梁收费占 25%，两者加起来超过 70%，剩下的人工成本和车险损失费都低于 30%。

国家邮政局于 2023 年第三季度发文公布了《快递服务满意度调查》的及时限时率测试结果。用户对快递服务的公众满意度得分在 2023 年第三季度为 82.9 分，同比上升 0.7 分。测试发现，整个快递服务流程的时限为 54.24 小时，较 2023 年第三季度缩短了 2.50 小时。72 小时准时率为 83.75%，比去年同期提高了 3.36 个百分点。

从物流运输成本看，数据显示，中国物流成本目前处于高位。尽管国内物流需求规模持续增长，但与发达国家相比，我国物流成本占比较高。特别是在运输成本构成中，燃油费和公路、桥梁收费占据了较大比例，表明了在成本管控方面还有很大的改进空间。因此，有必要采取有效的措施，如提高运输效率、降低燃油消耗、优化物流路线等，来降低物流运输成本。

物流行业需要关注的一个重要指标也是客户满意度。虽然市民对快递服务满意度的打分有所提升，但优化的空间依然存在。例如，在时限准时率测试中，虽然准时率有所提升，但仍有待进一步提高。这需要从提升服务质量、优化配送流程等方面着手，以提升客户的整体满意度。

1.2 研究意义

本课题的研究能在以下方面为物流行业优化成本和提高客户满意度：

1. 减少物流的运输成本：本项目的一个优化目标是降低总的配送总成本最低，且考虑了实时路况，时变车速等现实因素。有助于真实的减少运输成本。
2. 减少调度员的工作量：本项目能帮助调度员完成日常调度任务，减少调度员工作量，能有效提升调度员的幸福感，降低企业的成本。
3. 提高客户平均满意度：本项目的第二个优化目标为客户平均满意度，且由于降低了成本，考虑了时间窗因素，所以能有效降低客户的支出，因此能提高客户的平均满意度。

综上所述，本课题的研究有助于降低物流成本，提高客户满意度，因此研究具有现实意义和价值。

1.3 国内外研究现状

1.3.1 车辆路径问题

关于VRP问题的研究成果十分丰富，国内外总共上万篇相关文章，且由于其现实意义，大量企业投入了许多资源研究该问题。

VRP问题的解法目前主要有两大类，分别是精确求解方法，启发式算法。其中精确求解方法主要有分支界限法，分支界限定价法，列生成算法。启发式算法则更多，如基因算法，仿真退火，禁忌查找，蚁群算法，变域查找等等。由于深度学习近几年在其他领域取得了很多成果，部分学者也使用深度学习方法来求解VRP问题及其变体。

分支定界法[2]是一种求解整数规划问题的常用算法，可用于解决纯整数规划和混合整数规划问题。该方法通过将所有可行的解空间反复分割成越来越小的子集(分支)，并在每个子集(针对最小值的问题)的下界，对其目标进行搜索和迭代，从而计算出解集的各个子集。在每次分枝后，当发现某些子集的界限超出已知可行解集的目标值时，便不再进一步分枝，而是进行剪枝操作。这种方法的主要思路是不断地分枝和定界，以逐步缩小搜索空间，从而提高求解效率。

由于 TSP 是 VRP 的特例，因此 VRP 也属于 NP-HARD 问题，因此目前无法在多项式时间内得到最优的解法，因此精确算法只能在小范围的实例中求解，而大范围的实例则普遍使用启发式算法来求解。除了运行时间的差距，精确求解算法难度很大，实现十分复杂，而启发式算法一般算法思想简单，实现方便，因此启发式算法受到广泛的使用。

在运输管理中，VRP有很高的经济价值和意义，因此，很多研究人员一直致力于解决该问题。1959年，Dantzig[1]提出了CVRP问题。1987 年，SOLOMON 6 在 CVRP 即 CVRPTW 问题的基础上加入了时间窗口约束，并给出了一组至今仍在大量使用的被称为 SOLOMON 算例的算例。1999年，Laporte[7]提出了VRP问题的精确算法，同时Larsen[10]使用了Dantzig-Wolfe分解（DW分解算法）的精确方法来解决 CVRPTW。Lysgaard[11]等人提出了分支和切割的算法。

由于VRP是NP-hard问题，因此许多研究者使用启发式算法解决这个问题。Toth[14]在2003年提出了基于受限邻域概念的颗粒禁忌搜索策略。Golden[12]在1998年以及Wasil[13]在2005年结合了记录到记录的旅行和可变长度邻域列表，发现了许多新的最优解。Mester[15]在2005年提出了主动引导进化策略AGES并获得了许多最优解。这部分是因为使用了高质量的初始解决方案。其中VNS算法是Imran和Wassan[16]在2009年提出的算法，被认为是解决VRP问题最成功的元启发式算法之一。变邻域搜索算法的改进版本于 2007 年由Bräysy提出。Polacek[19]在2004年，以及Polat[20]在2015年也提出了使用VNS算法解决VRP问题。

除了精确算法和启发式算法，部分学者也采用深度学习的方式来求解VRP问题。Elias Khalil[22]在2017年使用图嵌入结构和深度学习方法求解VRP问题。根据结果，该方法在部分数据集上略差于启发式算法，但是在真实世界表现略好。Chaitanya[23]和Alex Nowak[24]在2017年以有监督的方式训练了一个图卷积网络（GCN）。Michel Deudon[27]在2018年将Irwan Bello[28]中的长短期记忆（LSTM）架构替换为Transformer架构，实现了更有效的学习方法。

综上，国内外已有多种探索，并有丰富的研究成果可供参考，如对汽车路径问题的分类、造型和求解方法等。且为了贴合实际，VRP问题的各类变体提出的速度越来越快。

1.3.2 粒子群算法

粒子群算法（Particle Swarm Optimization，PSO）是一种群体智能优化算法，灵感来源于鸟群或鱼群等生物群体的行为。该算法于 1995 年由美国的詹姆斯肯尼迪和鲁塞尔艾伯哈特提出，用于解决优化问题。

粒子群算法模拟了生物个体在搜索过程中，如鸟群或鱼群等的群体行为。在 PSO 算法中，在求解空间中沿着搜索方向移动的每一个求解空间潜在的解被称为粒子(particle)，其移动的方向和速度是根据它们的个体历史最优解法和群体历史最优解法(particle)调整的。

PSO 算法的基本思想是在空间中寻找最优的解法，通过模拟群体间的协作和信息共享。每个粒子都有定位向量和速度向量，这些位置向量通过迭代更新来搜索和解决空间(space)。粒子的运动在一定程度上受到随机性的影响，同时也受到个体历史最优解和全局历史最优解的影响。

粒子群演演算法的理论研究是对演算法本身进行探索和改进。SHI 等将惯性权重加入到粒子群算法的速度更新中，并通过大量实验检验了惯性权重对粒子群算法的求优能力的影响。实验结果表明，惯性权重的大小直接决定了整个粒子群在寻找优势过程中的策略，因为粒子跳出了局部最优和扩大了搜索范围。通过研究学习因子 C1 和 C2，Chaturvedi 等人得出算法可以在 [0，2] 时得到学习因子值的更好结果。Jin 等人提出了一种基于健康度的粒子群算法(HPSO)，通过对每一代中的异常粒子进行变异操作，通过对每一种粒子的健康度数计算，区分出正常粒子和异常粒子，从而改善其个体的最优值(ertification)。实验结果也表明，HPSO 拥有更好的搜索能力。

由于 PSO 存在容易陷入局部性最好理解的弊病，所以一些研究者也将研究的重点放在了如何提升 PSO 的整体搜寻能力上。综合学习粒子群算法是由梁等人提出的，其特点是在综合了整个种群粒子的历史最优信息时，更新单个粒子的飞行速度。因此，该算法可以保持良好的种群多样性，有效避免算法过早地陷入局部最优解法(particalpresident)的问题。Zhan 29 等人提出了具有快速收敛的全局搜索能力的自适应粒子群优化算法(APSO)。该算法实时计算并根据种群的空间分布信息和粒子的优缺点来确定粒子群的演化状态，并对算法中的各种参数根据种群的不同状态进行动态调整。这样的设计可以让算法的搜索表现更好，收敛的速度也更快。此外，APSO 还引入了有助于算法从局部最优解法中跳出来的精英学习策略。Maurice Clerc[31]在2004年提出了NoHope/ReHope机制，用于在种群陷入局部最优解时重新启动种群，此外，Maurice Clerc[31]还定义了一组行为，将PSO在解空间的连续变化映射为离散变化，并求解了TSP问题。粒子群优化算法夏学文等提出了粒子群优化算法，具有反向学习和局部学习的能力。这个算法记录了每一次迭代中，每个个体的最差粒子和历史最差位置的位置。一些粒子在算法停滞的情况下，为了让粒子摆脱局部最优，使用反向学习策略来学习这些较差的信息。同时，利用种群中较好的微粒位置，引导微粒进行局部的精细搜寻，使得算法的局部搜寻能力得到有效提升。

综上所述，国内外对于粒子群算法进行了各种各样的研究，研究成果十分丰富，对于解决相关问题具有很大的作用，但是对于理论上的研究还是略显欠缺。

1.3.3 粒子群算法在车辆路径问题中的应用

作为 NPH 问题的 VRP 问题，它的高难度解法是精确的，因此解决这类问题通常采用 PSO 等启发式算法。Gan 等人在汽车路径问题中引入碳排放因子，提出新的 VRP 模型，用 CLPSO 求解，并对比 GA 求解的结果。实验显示，CLPSO 算法的解法效果更好。Okulewicz 等人提出了一种两阶段的多种群 PSO 算法，并将其应用于动态 VRP 问题的解决， 并将其与其他最新提出的改进 PSO 算法在标准测试函数上进行了比较，结果均值和收敛速度在文章中提出的算法中均优于其他算法。Anisul[34]等人使用PSO混合VNS算法，用于求解CluVRP问题，在大量实例上取得良好的效果。

由此看来，PSO 解决 VRP 问题是很有潜力的。

1.4 研究内容和方法

1.4.1 研究内容

该研究将实际调度车辆的问题抽象为 VRP 车型，并使用 PSO 来解决这一问题。

PSO思想简单，易于实现，不需要优化函数满足连续、可微、可导等条件，收敛速度快，在大部分连续优化问题上都有良好的表现，但是对于离散优化问题并不能直接求解，搜索精度不高，而且存在容易陷入局部最优解的情况。

针对以上情况，本文采用以下方法对算法进行改进，并对其有效性进行模拟实验验证。现将具体调研情况说明如下：

（1）总结已有的研究：

通过广泛阅读相关文献，调研业界现状，对VRP问题和PSO算法的研究现状进行了总结，并且对PSO算法的特点和PSO算法在VRP问题方面的运用进行了阐述和概况。

（2）改进PSO算法与仿真实验：

针对PSO算法针对离散问题难以直接求解，搜索精度不高，容易陷入局部最优解的缺点，本研究使用了多种方式解决。对于VRP的离散状态编码，采用多维向量，将车辆和站点的对应关系、同一路线中站点访问顺序进行编码。针对 PSO 算法搜索精度不高、容易陷入局部最优解的问题，采用多种方式改进算法，包括混沌映射算法初始化种群、改善种群多样性、使用莱维飞行(LevyFlight)增强搜索能力、最后使用 kopt 算法改善解决质量等。最后通过高德地图的城市拥堵数据、Homberger数据集验证解的有效性，并与自适应粒子群算法进行比较，验证上述方法的有效性。

1.4.2 研究方法

研究方法主要有两大部分，即文献分析和模拟实验，研究方法主要有以下几种：

1. 文献分析法：

了解 VRP 问题和 PSO 算法的研究现状，通过阅读相关文献，总结分析 VRP 问题和 PSO 算法的原理、特征和改进方向，对行业现状进行研究的方式。

1. 数学建模法：

将考虑实时路况、时间窗约束、时变车速等多种因素，配送总成本最低、客户平均满意度最高的需求，定量分析问题，将原始问题抽象为VRP模型，并根据需求提到的非基本模型包含的约束和优化目标添加到模型中，便于理解和求解问题。

1. 仿真实验法：

通过Python程序在高德城市拥堵数据、Homberger数据集上仿真计算，得到结果并进行比较分析，得出结论。

1.5 论文组织结构

本文一共分为4个章节。第1章为绪论，主要介绍了课题背景、车辆路径问题和粒子群算法的概念、国内外研究现状以及本文的工作。第2章以相关算法和本文使用的技术方案为重点，以研究内容和相关技术方案为主。第3章是设计方案与实验细节，主要介绍了算法实现、参数调节、实验结果与分析的内容。第4章总括与展望，概括全文，展望未来。

1. 主要研究内容和相关技术方案

2.1 车辆路径问题

2.1.1 车辆路径问题介绍

车辆路径问题(VRP，VehicleRoutingProblem)，是指在给定一组客户需求点和一组有容量限制的车辆的情况下，确定每辆车的行驶路线，从而在最小程度上降低行驶距离或总成本的同时满足所有客户的需求。VRP 是在物流、运输、配送等领域广泛应用的具有重要理论和实际应用价值的经典组合优化问题。

VRP问题通常是一个NP-hard问题，难以通过穷举法求解，因为随着客户数量的增加，搜索空间呈指数增长。为了解决VRP问题，研究者提出了许多启发式算法和元启发式算法，如遗传算法、蚁群算法、模拟退火算法、粒子群算法等。这些算法通过在可行解空间中进行搜索，并通过适当的策略来选择和调整解决方案，以寻找较优解。

VRP 问题广泛应用于现实生活中，如物流配送、城市交通规划、公交优化等领域。通过优化车辆的路线和调度安排，可以有效地降低运输成本、提高运输效率，从而实现资源的合理利用和社会经济效益的最大化。

VRP问题通常包含以下要素：

1. 客户点：即客户需要送达的地点。
2. 用车：每辆车只到一个客户点访问一次，服务完的车辆每辆只有一辆。
3. 优化目标：最小化的目标函数。通常为服务总成本或者车辆行驶距离。

除了上述要素，根据实际问题的不同，还会添加或者修改约束和优化目标。根据常见的现实要求，VRP问题有以下几种常见变体。容量限制车辆路径问题（CVRP），添加了客户需求量和车辆容积的条件。时间窗车辆路径问题（VRPTW），添加了客户点仅在特定时间窗接受车辆访问的条件。混合车辆路径问题，即车辆不再是同质车辆，存在容量限制、车辆速度、运营成本等不同类型的车辆。

2.1.2 车辆路径问题的数学建模

本文研究的对象，是带时间窗的容积限制时间依赖多目标车辆路径问题。

其中时间依赖体现在现实中不同时间段路段有不同的拥堵系数，所以车辆到达目标客户点需要的时间也不一样。

本文的研究对象包含两个优化目标，分别是配送总成本最低和客户平均满意度最高。其中配送费主要来自于车辆行驶过程中所消耗的油费和驾驶员的工资，由于驾驶员的工资一般由基本工资和绩效两部分构成，因此综合考虑，可以将行驶距离作为配送费的最优化功能。客户满意度和多个因素相关，例如准时率、服务态度、收费标准等，其中准时率是影响客户满意度的最大因素，所以将超时罚函数的和作为客户平均满意度的优化函数。

本文研究目标的假设条件为：

1. 客户数量确定，位置确定且需求量确定，。
2. 配送车辆数目确定，且有容积限制，且载货量不能超过容积限制。
3. 客户点之间的距离已知，且道路拥堵情况符合历史规律。
4. 货物在路途上的损耗不计入成本。
5. 如果车辆没有在规定的时间段内到达顾客点，顾客的满意度就会下降，且随着与软时窗的差距越来越大。
6. 每个客户点有且仅有一辆车在第一次到达时完成服务。
7. 车辆从仓库出发，回到仓库。

计算车辆到达时间需要特殊处理，方便程序实现和提到代码运行效率。在无拥堵车辆速度恒定为v的情况下，在时间t1，从a点出发，到达b点，计算到达时间t2，设距离为L，拥堵系数函数为tc（Traffic Congestion），可以视为求解下面积分方程：

（2-1）

从方程（2-1）可以看出，设TC=，则方程（2-1）被转化为

（2-2）

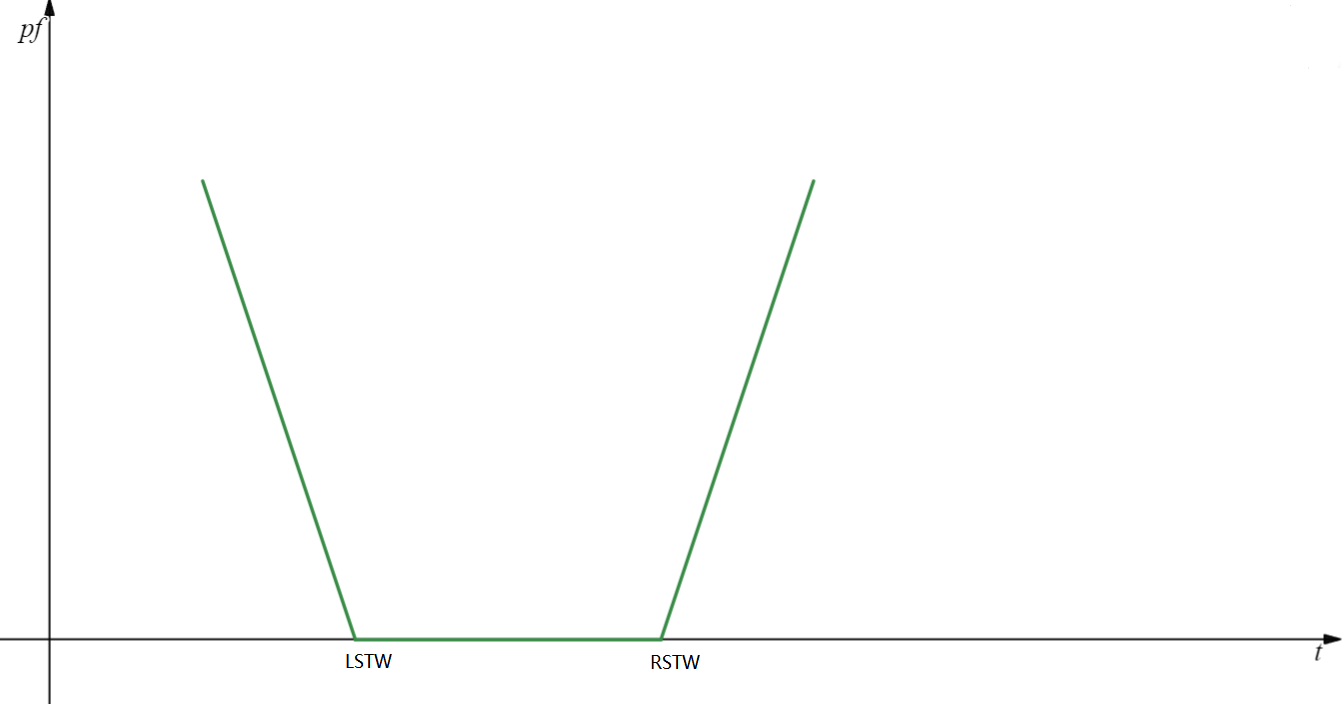
则t2可以通过查表获取，因为TC显然为非递减函数，所以查表时间复杂度为O（log n），在分钟精度下的交通拥堵情况，可以认为TC的反函数ITC以常数时间运行。

根据调研和文献阅读，不同的行业、不同货物业务种类在非指定时间送达造成的损失有着很大的差别。通用的损失有人力成本上升带来的。部分行业提早到来导致的损失要大于晚到的损失，如冷藏货物仓储成本较高。有的晚到的行业，比如急送业务，比起早到的行业，亏损更大。

综上所述，本文设早到和晚到的损失一致，因此超时的罚函数pf（Penalty Function）设为如下函数：

（2-3）

函数图像如下所示。



其中STW为软时间窗（Soft Time Window），LSTW为时间戳的左边界，RSTW为时间窗的右边界。

为了便于讨论，我们将车辆路线的罚函数也定义为pf，但是含义为路线上的每个客户点的罚函数之和。

下面给出本题考查目标的形式化表述。

设给定n个客户点，m辆车，每个客户点都有软时间窗（Soft Time Window，STW），服务需求量r，每辆车辆的容积为q，无堵塞车辆速度为v，两客户点之间的距离函数为w，拥堵系数函数为tc，tc原函数的反函数为ITC，不在指定时间服务的罚函数pf。

优化目标：

（2-4）

（2-5）

约束：

（2-6）

（2-7）

（2-8）

（2-9）

（2-10）

其中公式(2-4)表示最小的车辆行驶距离，公式（2-5）表示最小化用户的不满意度，pf（k）表示计算车辆 k 服务的客户点处罚函数之和，公式(2-6)表示每个客户点都有且仅有一辆车服务，公式(2-7)表示所有车辆从起点出发，公式(2-8)表示所有车辆返回终点表示全部车辆，公式(2-9)不超过容积的载货能力，公式(2-10)为决策变量，为1表示车辆k由i点至j点。

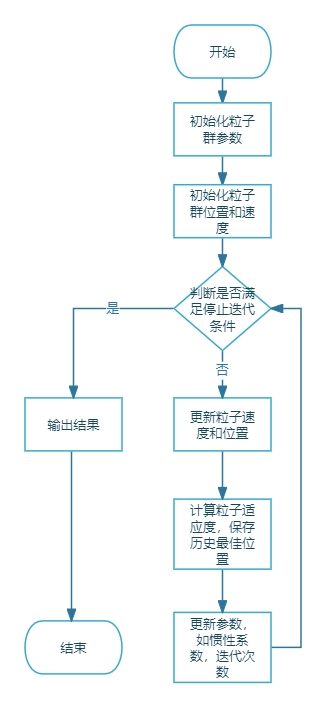
2.2 粒子群算法

2.2.1 粒子群算法流程

粒子群算法(PSO)是一种仿生算法，由杰梅斯肯内迪(JamesKennedy)和鲁塞尔厄伯哈特(RussellEberhart)通过观察鸟类群体觅食而建立。粒子群算法是一种复杂适应系统(ComplexAdaptiveSystem，CAS)，复杂适应系统理论认为系统演化的动力本质上来自于系统内部， 微观主体之间的相互作用产生宏观上的复杂性现象，其研究思想集中于系统内部各要素之间的相互作用(complexadaptiveSystem)。

粒子群算法的基本原理是生成若干个粒子，称为一个种群，每个粒子都有位置向量和速度向量，位置表示一个可行解，速度表示位置变化率，粒子可以感知当前解的适应程度，粒子可以记录自己的历史最佳位置，感知到种群历史最佳位置，粒子可以记录自己的历史最佳位置，可以感知到一个种群的位置，每个种群并将速度向量改变，向最好的位置靠近，根据学习率、自身和种群粒子的最佳位置完成搜寻。

粒子群算法的流程可以这样表述：先初始化粒子群的个数、迭代的最大次数、惯性权重等粒子群参数，再初始化粒子，包括粒子的位置、速度等，然后计算适应程度，判断是否满足停止迭代的条件； 若符合，结果输出，过程结束；如果不满足，就更新粒子，把历史上最好的种群和粒子位置保存下来，计算适应程度，更新参数，再判断是否可以停止迭代。



2.2.2 粒子群编码与解码

由于粒子群算法无法直接处理离散数据的优化问题，我们采用以下编码解码方式将连续数据映射到离散数据中。

每个粒子位置都是由向量，构成的 n 维向量，每个粒子速度也是记为，的两个 n 维向量。其中的元素属于{1,2,3,...,m}，没有限制。向量的第i个元素为k表示客户点i由车辆k服务。对于所有由车辆k服务的客户点集合S，根据对应位置的元素排序，决定被车辆k服务的顺序。

由于后续算法实现需要将实际的方案转化为位置编码，所以对于一个车辆k的路径path，path为一个由{1,2,3,...,m}中的元素构成的数组，元素先后顺序表示车辆k服务客户点的顺序。构建向量，若是未将向量对应位置置为k即可。构建向量，按照path的顺序分别将向量对应位置按照未进行编码操作前的元素赋值即可，同理生成对应的速度向量。

2.2.3 多目标价值判断方式

由于这项研究的目标函数有两个，所以我们采用了多目标优化的粒子群算法。

常见的多目标优化决策方式有加权平均法，基于惩罚的边界交叉方式、帕累托前沿方式等。

其中加权平均法是指定每个目标值的权重，最后根据目标的加权平均值计算适应度，是最常见最易于实现的方法。基于惩罚的边界交叉方法的思想是允许解不在预设权重的向量上，但是需要接受一个惩罚值，以此保证解能被约束在预设权重的向量附近。帕累托前沿使用帕累托非支配解为解集，解集里的每一个解都满足不存在另一个解所有目标函数的值都不劣与这个解，有至少一个目标函数的值优于这个解。

本文使用帕累托最优的方式进行排序筛选粒子，最终返回一个帕累托前沿给用户，让用户自行决定选择哪一种方案。

构造帕累托最优解集有以下几种常用方法，这里介绍简单遍历法和快速排序法。简单遍历法构造帕累托解集，将每一个元素都与其他元素比较，当该元素不存在有其他元素支配该元素，则该元素成为非支配解，该算法的时间复杂度为O（）。快速排序法构造帕累托，仿造快速排序的思想，选择一个元素，将数组分为2部分，劣于该元素的元素，不劣于该元素的元素，然后对于每个部分递归排序，最终得到每个解的层次，取最上层即可，该算法使用了帕累托支配关系的传递性，时间复杂度也为O（），但是平均运行时间少于简单遍历法。

2.2.4 线程池

本研究使用Python的ThreadPoolExecutor作为线程池。

线程池核心参数包括最大线程数和最大等待队列数。由于本研究使用的程序为计算密集型的程序，所以最大线程数可以接近逻辑核心数，但是因为存在绘图，输出保存代际信息等IO操作，所以任然需要测试CPU耗时与IO耗时，测试结果为CPU耗时/IO耗时约为9.4，所以我们设置最大线程数为逻辑核心数\*1.2+1。由于本研究内存占用较小，预期等待任务数量较少，所以最大等待队列数设为无穷。

本研究能够使用多线程并行计算的部分为粒子分数计算，kopt优化。

2.3 改进粒子群的方案

虽然 PSO 的想法简单、容易实现、不需要很好的性质优化功能和快速的收敛功能，但存在容易陷入局部最优解法，因此离散优化问题不能直接解决、查找准确性不高等问题。针对这些情况，本研究采用以下方案提升算法的搜索能力。

2.3.1 混沌映射算法

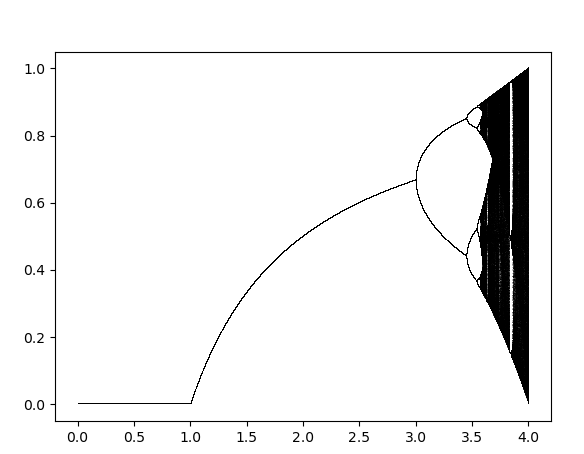
混沌映射算法常用于生成随机数，在优化领域，经过混沌映射生成的初始种群通常由更好的种群多样性。

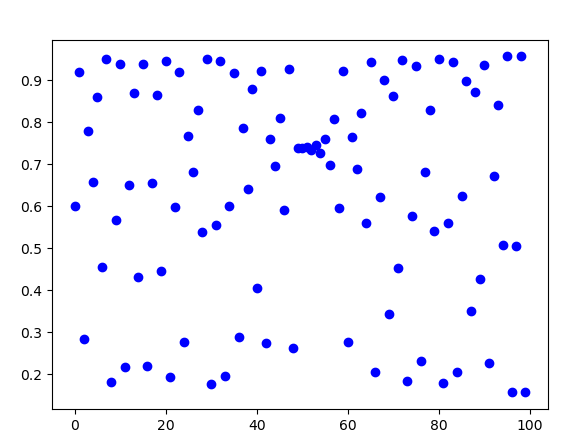
虽然混沌映射算法能提高初始种群多样性，但是经过实验，对于最终结果无明显提升，仅在初期加快了算法收敛的速率。

常见的混沌映射算法有下面几种：

1. Logistic映射：又称虫口映射，是一种简单的二次混沌映射算法，具体生成公式如下，其中μ∈[0,4],当μ在3.6到4之间时，系统出现混沌现象,具体生成公式和样例如下。

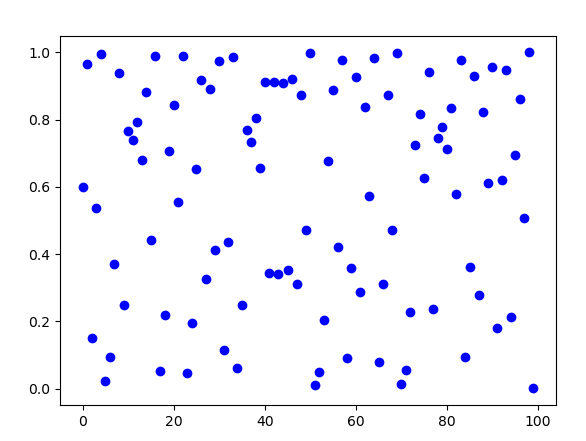
（2-10）





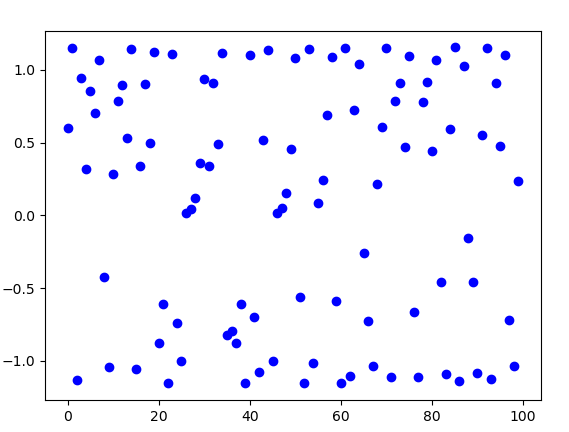
1. logisticsSine映射：与Logistic映射类似，但是添加了一项与正弦函数相关的项，具体生成公式和样例如下。

（2-11）



1. Cubic映射：与逻辑混沌映射相比，Cubic映射具有更好的混沌遍历能力。它的优化速度快、精度高，具体生成公式和样例如下。

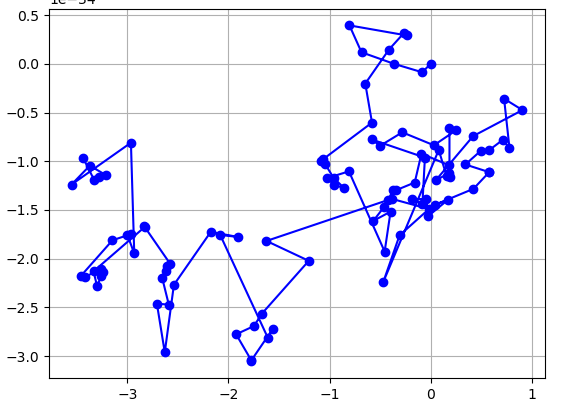
（2-12）

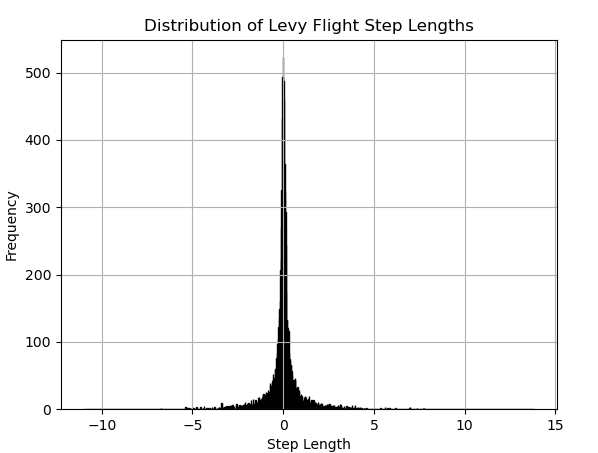


具体的粒子群初始化过程可以描述为：首先生成4n个混沌映射随机数，第1到n个数分为一组，归一化后乘以m，再赋值给向量，其余的3n个数字依次分配给，，向量。

2.3.2 Levy飞行

莱维飞行(Levyflight)得名于法国数学家保罗莱维(PaulLevy)，是指步长概率分布为重尾分布的随机行走，即在随机行走过程中出现大跨步的概率相对较高。莱维飞行属于马尔可夫过程。对一般的类幂律的步长分布，经过很多步之后与起始点的距离的分布将会因一般化的中心极限定理而趋近于稳定分布（莱维α稳定分布）。所以，莱维飞行可以描述很多随机行走的过程。下面给出一副莱维飞行的样例图和分布图。





可以看出莱维飞行在大部分时候都是小步随机行走，偶尔出现大步的行走。

莱维飞行的分布函数比较复杂，一般采用曼特格娜(Mantegna)的方法生成，具体如下：

（2-12）

（2-13）

（2-14）

（2-15）

其中β为控制随机变量分布的参数。

由于莱维飞行可能生成很大的数字，所以采用对数函数减小数字的大小，防止出现粒子完全跳出搜索范围，浪费学习次数，公式如下：

（2-16）

该研究将莱维飞行用于增强粒子历史最优位置的粒子学习和群体历史最优位置的学习。具体改进公式如下：

（2-16）

其中c为c1或c2，s为莱维飞行生成的随机变量的值，为个体历史最优位置或群体历史最优位置与当前位置的差值。

2.3.3 kopt算子

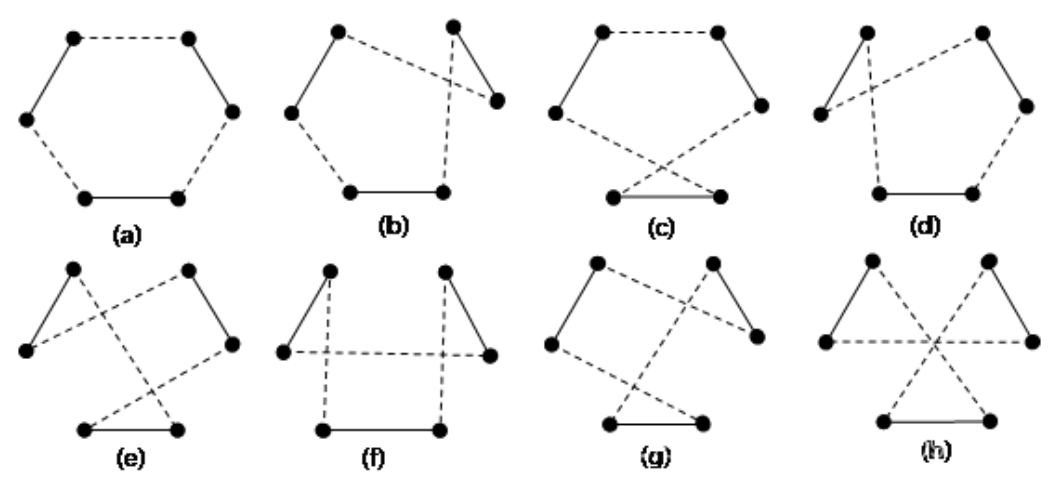
由于粒子群算法无法针对某一个解空间领域进行精细的搜索，所以我们变异算子来对一个较优解进行精细的搜索，以得到质量更佳的解。

常见的路径变异算子分为路径间变异算子和路径内变异算子，路径间变异算子有交换转移方法、2opt\*等方法，路径内变异算子有2opt、3opt、or-opt等等。

交换转移方法是在两个路径中各自选择一个客户点，然后交换，并生成对应的边，完成变异。2opt\*方法和交换转移方法的不同之处在于2opt\*方法不是交换客户点，而是各自选择客户点，然后将生成的4条子路径按照原有的行驶顺序两两交换，形成新的路径。

2opt与3opt是kopt的实例，kopt是一个路径内变异算子，是将一个路径移除k条边，然后增加k条边生成新的路径的方式。对于新路径，如果新路径比原来更好，就保留新路径，把原来的路径去掉。kopt包括原先的情况一共有种情况。

以3opt为例，我们可以看到，一共有8种情况。



由于kopt是路径内变异算子，所以对于每辆车辆预期的车辆路线，都能单独进行kopt变异，也因为本研究使用构造帕累托前沿的方式返回最优解集，所以每个车辆路径都存在多个优化结果，所有的车辆一起组合构造帕累托最优解集的时间复杂度是指数级别的时间复杂度，在大规模用例下难以完成优化，为了能得到更优的结果，我们采用剪枝的方式减少中间生成的解集。具体的剪枝方式为对于每一次合并两条路径，我们将生成的结果种的支配解删除，留下非支配解继续生成解集，因为支配解生成的解会被非支配解与相同的解合并的解支配，所以这一步不会影响最终结果。另一个剪枝策略是限制合并路径时最大的解集数量，这里我们采用多次排序，分别取每一个目标值最优的一部分解继续合并生成帕累托最优解，这种方式保证了种群的多样性。

为了提升学习效率，我们将kopt生成的路线使用上文提到的反编码方式重新编码为粒子，并作为粒子群的全局历史最优位置保存。

由于对于一条路径使用kopt代价很大，所以我们仅针对在自然生成的帕累托前沿的解进行kopt变异优化，并且设置了最大的优化次数。

算法伪代码如下：

Begin

传入解

路径信息数组=解的解码

车辆路径优化结果集=[]

for 车辆路径 in 路径信息数组：

路径分段数组=车辆路径的随机切割

路径分段数组=路径分段数组的全排列

路径分段数组=路径分段数组的随机翻转

for 组合结果 in 路径分段数组：

计算消耗与满意度

路径分段数组=非支配集（路径分段数组）

车辆路径优化结果集+=路径分段数组

车辆路径优化组合=[]

for 车辆路径优化结果 in 车辆路径优化结果集：

车辆路径优化组合\*=车辆路径优化结果

车辆路径优化组合=非支配集（车辆路径优化组合）

车辆路径优化组合=车辆路径优化组合.sublist(0,最大搜索宽 度)

return 车辆路径优化组合

End

2.4 对照算法

为了体现本文使用的改进方法的有效性，本文将把本文改进的PSO算法与自适应粒子群算法进行比较。

自适应粒子群算法使用进化状态评估、系统自适应参数和精英学习策略解决PSO算法的收敛速度慢与容易陷入局部最优解的问题。

2.4.1 进化状态评估

进化状态评估使用欧氏距离来计算一个粒子与其他粒子的平均距离，公式如下：

（2-17）

其中 N 为粒子群体的种群数量，D 为维度大小的粒子。

然后选择最优的值为，计算在归一化后的数值f记为进化因子，即

（2-18）

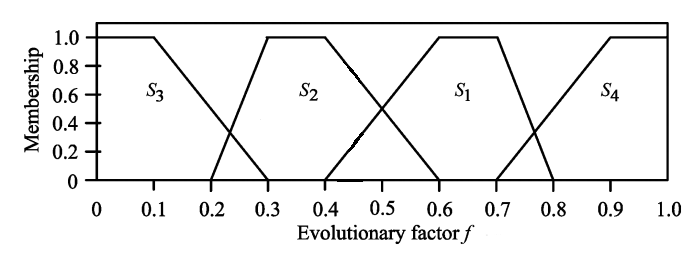
参考资料显示，该系统共有四种状态，分别为 S1(探索)、S2(发现)、S3(收敛)和 S4(跳出)，其对应的隶属函数和分布图如下：

（2-19）

（2-20）

（2-21）

（2-22）



可以看到部分区域存在重叠，重叠部分有两种划分归属的方式，一种是按照隶属度大小来划分状态即隶属度大的为被选择的状态，另一种是设置状态转移规则，即指定状态之间的转移关系，例如当前f=0.55，可以看到S1=0.75，S2=0.25，S1>S2，上一个状态为S2，则为了状态稳定性，不能设定当前状态为S2。这里我们选择直接按照隶属度来完成状态划分。

2.4.2 系统自适应参数

PSO中存在三个关键参数，分别是w、c1、c2，其意义分别为粒子的惯性、粒子对个体经验的学习率、粒子对群体经验的学习率，根据经验通常设w=0.7，c1=c2=1.4。

在自适应粒子群算法中，这三个参数会随着系统的进化而改变。

惯性权重w在算法开始时应当大，在算法将要结束时应当减小，以便增强搜索能力和收敛速度，所以w和f的关系如下：

（2-23）

为了增加粒子的探索能力，防止陷入局部最优解，在探索阶段增加c1，减少c2，增强种群的多样性；在发现阶段轻微增加c1，轻微减少c2，既能避免粒子群种群多样性丧失，又能增加粒子群的局部搜索能力；在收敛状态轻微增加c1，轻增加c2，能减少粒子惯性的影响，增加收敛速度，并且防止陷入局部最优解；在跳出阶段减少c1，增加c2，能快速收敛到当前的全局历史最优解。

我们使用随机值r来改变c，并且将每一代之间|Δc|控制在0.1以内，为了表示轻微增加和减少，使用0.5r来改变c，如果c1与c2之和超过了4，则使用c1和c2的归一化后的值作为新的c1和c2。

（2-24）

2.4.3 精英学习策略

精英学习策略采用部分学习的方式来帮助全局最优粒子跳出局部最优区域，向一个可能更好的区域演化。

精英学习策略把目标的其中随机一个维度进行变化，设维度为d，加入高斯扰动，具体公式如下：

（2-25）

其中是学习率，随着粒子群的进化而线性下降，变化公式如下：

（2-26）

2.5 本章小结

本章介绍了车辆路径问题，并且给出了形式化的描述。然后介绍了多目标粒子群算法和多目标价值判断方式，选择使用构造帕累托前沿的方式完成价值判断，最大的保留算法结果信息。之后介绍了用于改进粒子群算法的混沌映射算法、莱维飞行和kopt算法，分别用于粒子的初始化，粒子学习过程的优化和粒子群算法结果的优化。最后介绍了用于做对照实验的自适应粒子群算法，介绍了自适应粒子群算法的进化状态评估、系统自适应参数和精英学习策略。

1. 设计方案和实验细节

3.1 实验环境

实验环境的系统为Windows 10专业版，处理器为AMD Ryzen 5 4600H with Radeon Graphics，机带RAM大小为16.0 GB (15.4 GB 可用)。Python环境为Python3.9.12。

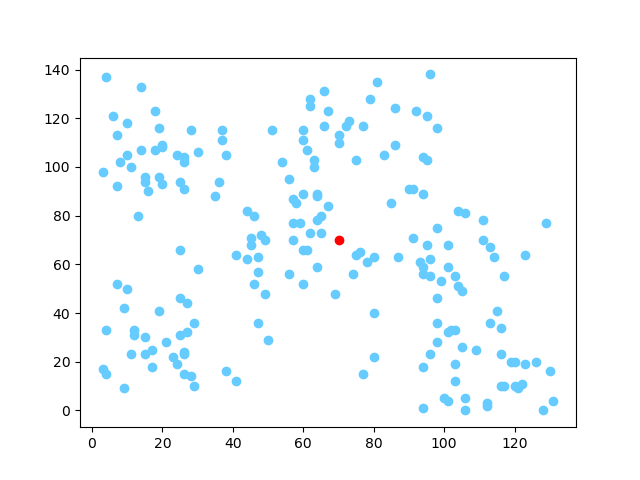
3.2 参数设置与数据集解释

本研究使用的模型使用以下参数，分别为粒子群迭代次数G=100次，粒子群大小N=150，惯性参数w=0.7，粒子学习率c1=c2=1.4，混沌映射算法选择logisticsSine算法，初始值seed=0.8和控制参数μ=3.831，莱维飞行控制参数β=0.5，kopt算法的分段数k=4。

拥堵数据集采用高德城市拥堵数据，收录了全国共计100个城市从2024-02-01到2024-04-15的拥堵延时指数与高延时比例，通过去除节假日等不符合日常工作条件的数据，得到全国城市拥堵数据的平均值，作为模拟真实道路的拥堵情况数据。

道路信息使用Homberger数据集，该数据集的每个实例都含有200个客户点，以及50辆车辆，每个实例的主要区别在于车辆容积与客户点的需求不一致。下面为一个实例的数据和客户点分布展示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 客户编号 | x坐标 | y坐标 | 需求量 | 开始时间 | 结束时间 |
| 0 | 70 | 70 | 0 | 0 | 2535 |
| 1 | 10 | 86 | 14 | 1402 | 2002 |
| 2 | 16 | 62 | 15 | 1344 | 1944 |
| 3 | 84 | 36 | 10 | 167 | 767 |
| 4 | 79 | 72 | 22 | 9 | 609 |
| 5 | 16 | 39 | 20 | 84 | 684 |
| 6 | 61 | 54 | 10 | 18 | 618 |
| 7 | 96 | 26 | 10 | 51 | 651 |
| 8 | 16 | 92 | 10 | 1195 | 1795 |
| 9 | 6 | 25 | 10 | 281 | 881 |
| 10 | 85 | 106 | 20 | 39 | 639 |
| 11 | 37 | 125 | 10 | 1134 | 1734 |
| 12 | 87 | 108 | 20 | 41 | 641 |
| 13 | 18 | 104 | 18 | 1107 | 1707 |
| 14 | 61 | 48 | 20 | 30 | 630 |
| 15 | 97 | 29 | 30 | 49 | 649 |
| ... | ... | ... | ... | ... | ... |
| 199 | 26 | 105 | 40 | 56 | 656 |
| 200 | 60 | 23 | 16 | 48 | 648 |



其中红色的点（编号为0的客户）为仓库，蓝色的点为客户点。

3.3 模型设计

3.3.1 算法流程

本研究算法的主要流程为：

输入：客户数据集与拥堵信息数据集。

输出：解集的帕累托前沿图与解集对应的路线。

步骤1：读取数据集，初始化粒子群参数。

步骤2：根据混沌映射方法的公式（2-11）初始化粒子位置和速度向量，计算初始适应度，保存历史最佳粒子信息。

步骤3：判断是否迭代完成，完成则跳到步骤8。

步骤4：根据莱维飞行的公式（2-15）、（2-16）更新粒子速度与粒子位置。

步骤5：计算粒子适应度，使用kopt方法优化解集，保存历史最佳粒子。

步骤6：更新迭代参数。

步骤7：返回步骤3。

步骤8：输出解集的帕累托前沿图与解集对应的路线。

3.3.2 算法伪代码

Begin

读取数据集

初始化参数

使用logisticsSine算法初始化种群

for i in range(0,G):

for 粒子 in 种群:

计算粒子花费和满意度

粒子历史最佳=粒子历史最佳+粒子

粒子历史最佳=非支配集（粒子历史最佳）

全局历史最佳=粒子群+全局历史最佳

全局历史最佳=非支配集（全局历史最佳）

for 最佳粒子 in 全局历史最佳:

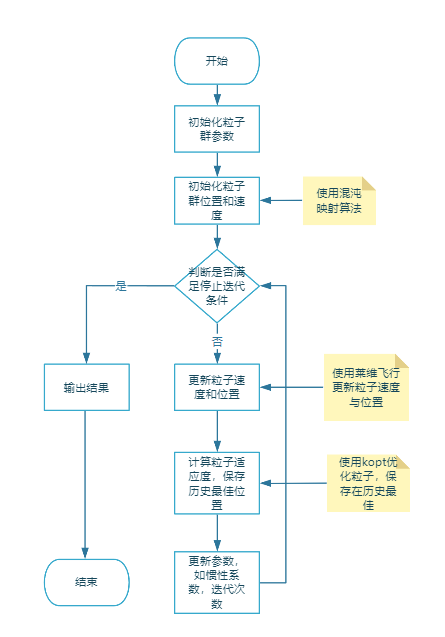
if(最佳粒子.kopt次数<最大kopt次数):

最佳粒子.kopt次数=最佳粒子.kopt次数+1

kopt集合=kopt集合+kopt(最佳粒子)

End

3.3.3 算法流程图

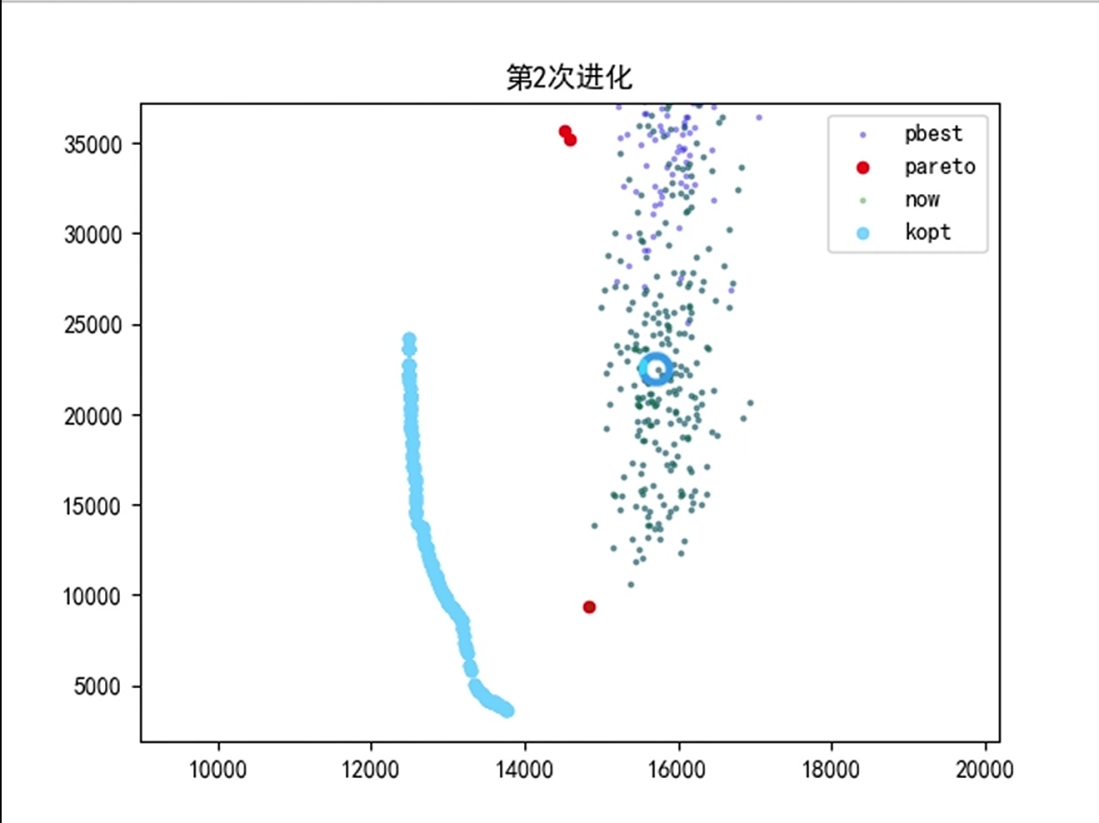
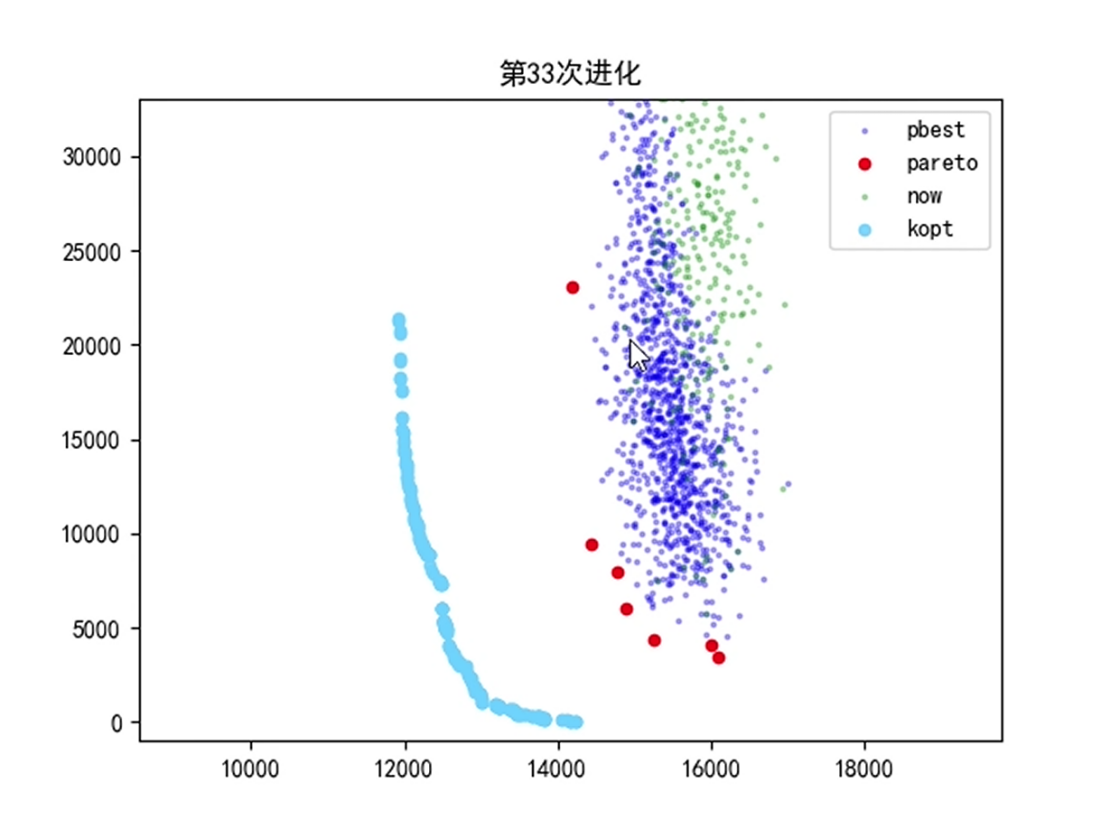


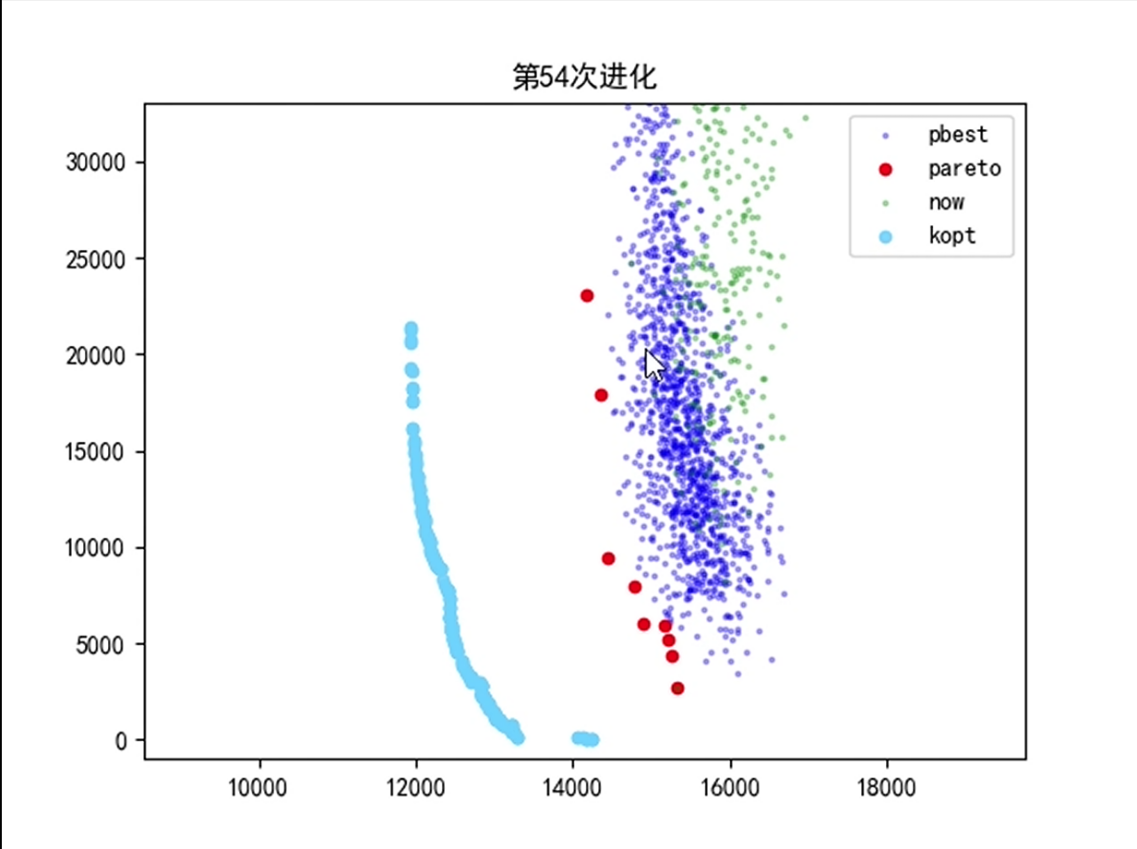
3.4 实验结果与分析

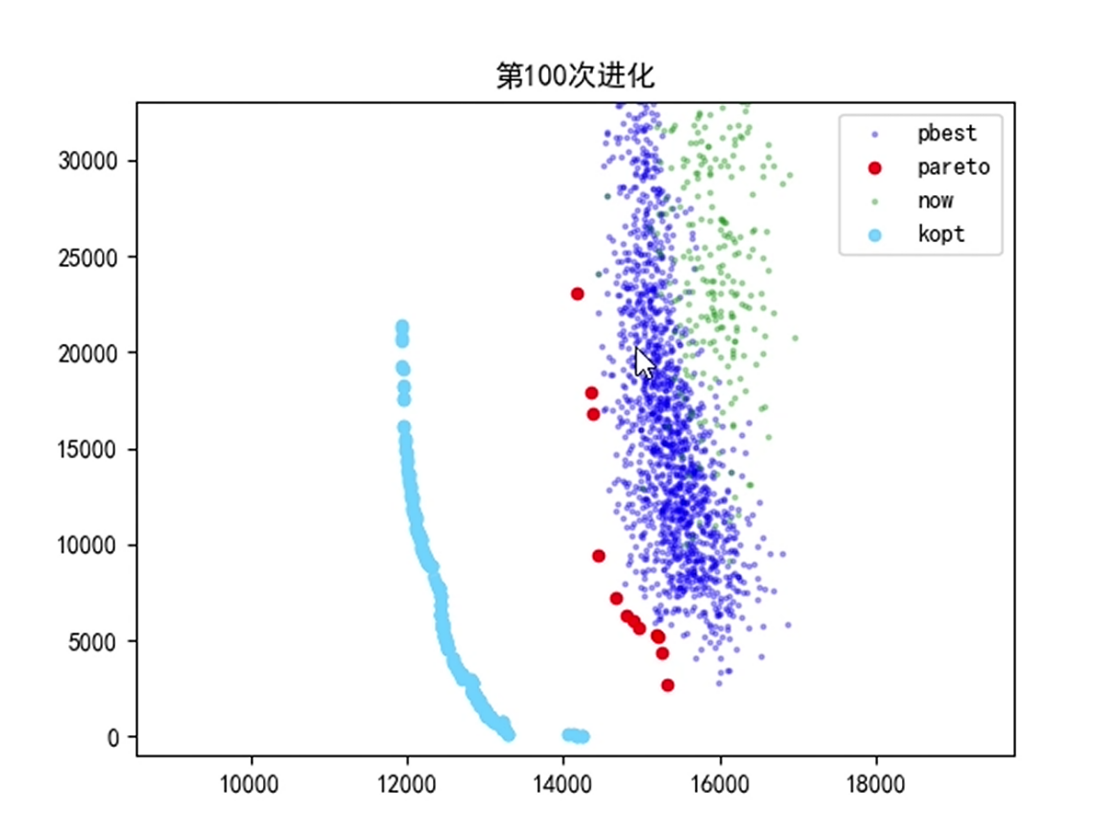
3.4.1 实验截图

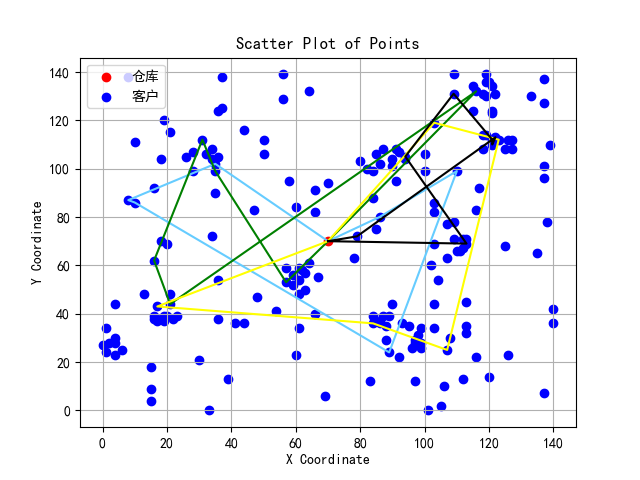
下文图为本文模型与自适应粒子群算法的帕累托前沿图与最终生成解集的一个实例的部分路线图，其中深蓝色点pbset是粒子历史最佳粒子，红色点pareto粒子群自然生成的种群历史最佳粒子，绿色点now是当前粒子的得分，浅蓝色点kopt为种群历史最佳经过kopt算法优化过的解集。

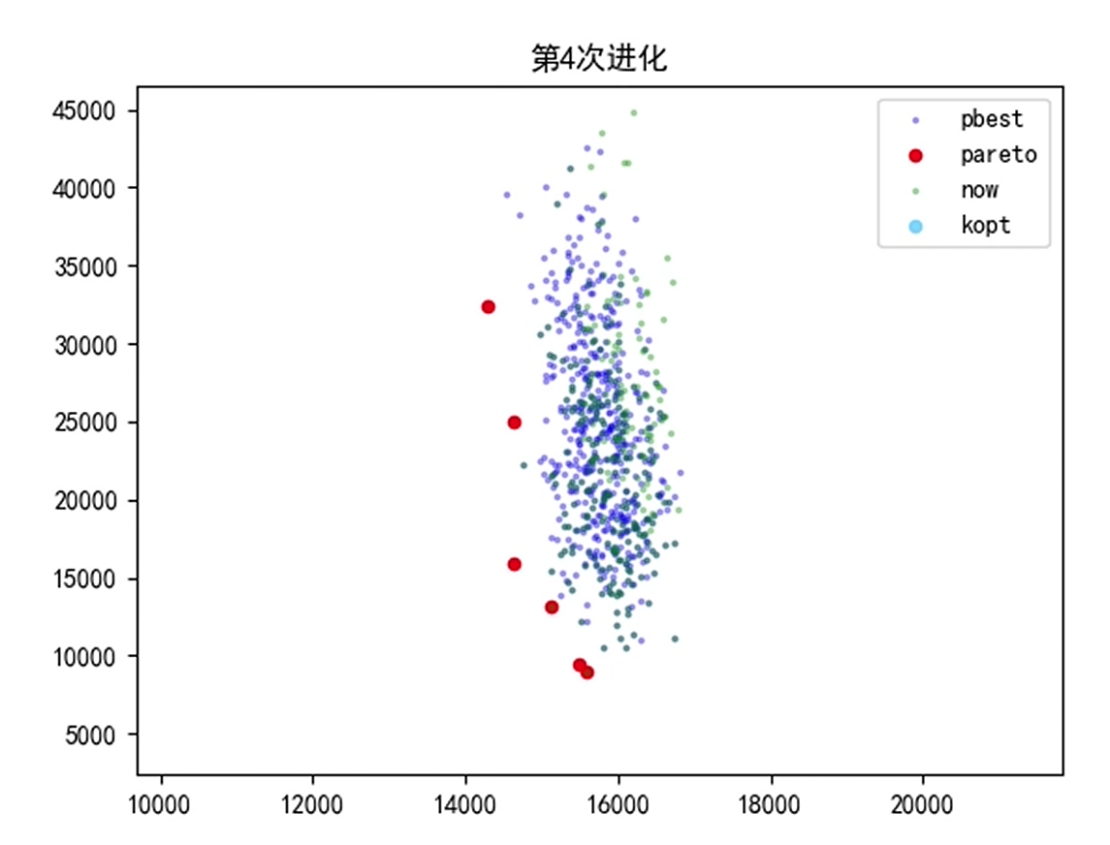
图像的横轴为成本，纵轴为时间罚函数之和。

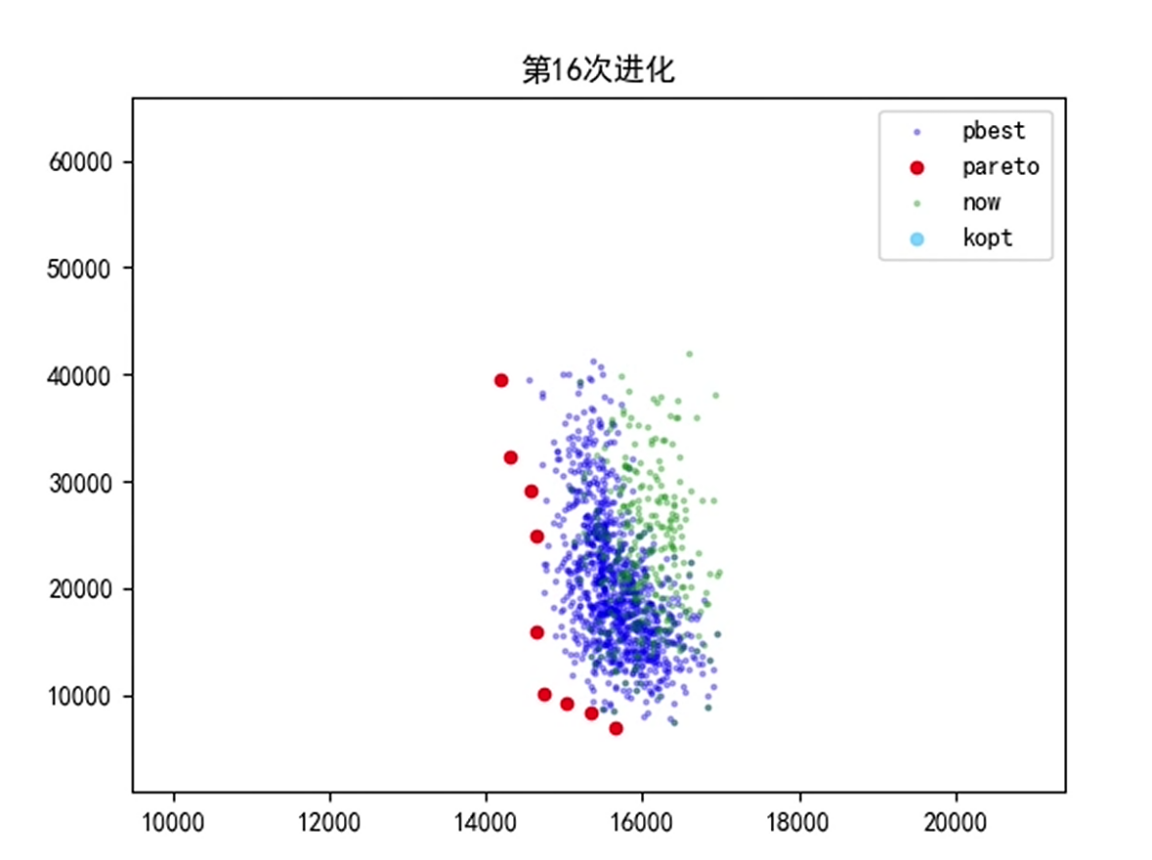
 

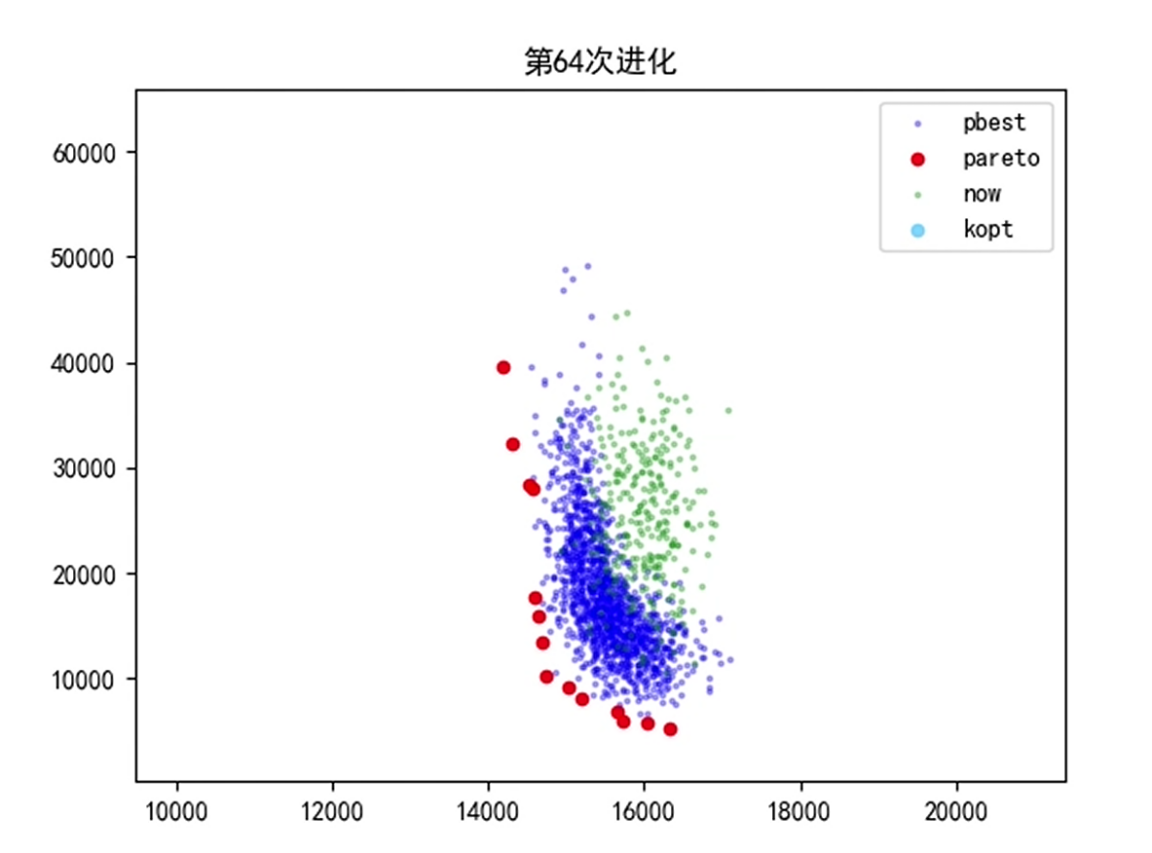


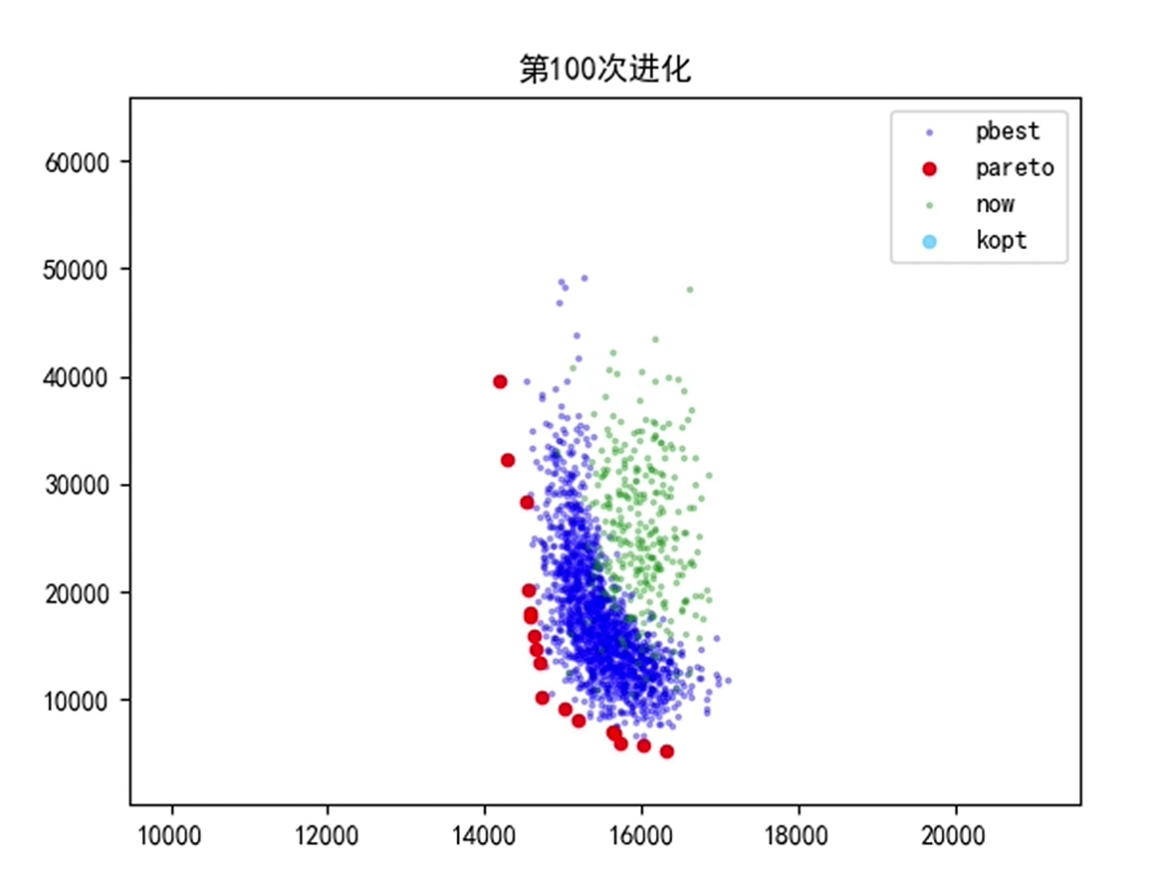


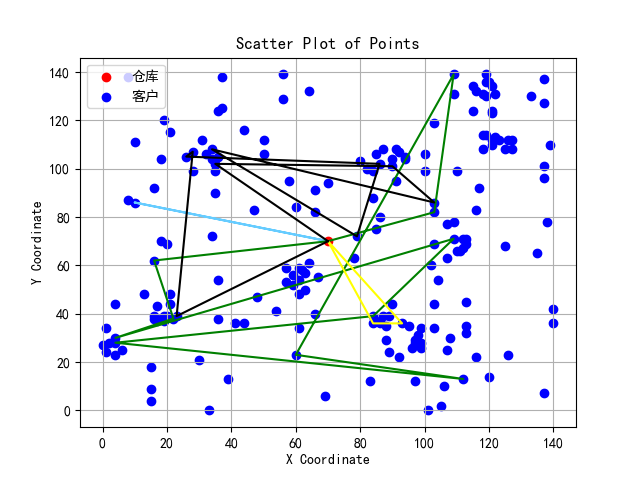












从图中可以看出，本文所使用的算法不论是kopt形成的帕累托前沿还是自然生成的帕累托前沿，都优于自适应粒子群的帕累托前沿，并且kopt形成的帕累托前沿明显优于本文算法自然生成的帕累托前沿。

3.5 本章小结

本章介绍了本实验的实验环境，包括物理环境和虚拟环境，然后介绍了实验参数和数据集，并展示了部分数据，之后介绍了算法的流程和代码，最后展示了实验截图，并与对照算法自适应粒子群进行比较。

1. 总结与展望

4.1 总结

本研究使用粒子群算法进行车辆调度能优化运输效率、节省成本、提高客户满意度、应对复杂情况和提高管理水平。通过算法的规划和智能调度，可以确保货物及时送达目的地，减少运输成本和时间浪费，提升客户满意度，有效应对交通堵塞和突发事件，提高管理水平和运营效率。

本文使用构造帕累托前沿的方式解决多目标车辆路径问题，使用多线程技术加速计算。针对粒子群算法的缺陷，本文使用了混沌映射方法提升种群的初始种群多样性，减少粒子群陷入局部最优的可能，使用莱维飞行增加粒子的学习能力，使用kopt优化提升解集的质量。

最后，本文介绍了实验环境的配置以及粒子群模型的主要参数设置，介绍了本文使用的数据集的来源并展示了部分数据，之后给出了模型的流程、伪代码以及流程图。然后展示了部分测试结果并于对比算法进行了比较，验证了算法的有效性。

4.2 展望

本文目前仅测试了3种混沌映射算法在粒子群初始化的表现，并且参数调节方面缺乏有效的方式，kopt算法时间复杂度过高，剪枝方式可能将最优解的部分删除，且除了kopt的路径优化方案，存在其他路径优化方案如路径间优化方案可以使用，并且数据集数据少，道路数据仅60个实例，难以验证算法的泛用性。针对上述问题，未来可以增加混沌映射算法的选择和测试，优化拓展路径优化算法，增加数据集大小。

致谢

四载恍然一瞬，回首即将毕业。四年里，不论是生活还是学习我都学到了许多全新的东西，也认识了许多很好的人。如今毕业之际，我想对所有曾经帮助过我的人表达感谢之情。

首先我想感谢的是中南大学和软件工程专业的诸位老师。在校四年，我能成长为现在这样的人离不开老师们的帮助。我们软件工程的老师，不仅完成教书的任务，还担当了育人的职责。在学习中，老师言传身教，不仅通过书本传授知识，也通过自己严谨的治学和人格魅力教学生如何成为更好的人。没有何老师的帮助，我无法

参考文献

[1] Dantzig, G.B. and Ramser, J.H. (1959) The Truck Dispatching Problem. Management Science, 6, 80-91.http://dx.doi.org/10.1287/mnsc.6.1.80

[2] Clerc, M. (2004). Discrete Particle Swarm Optimization, illustrated by the Traveling Salesman Problem. In: New Optimization Techniques in Engineering. Studies in Fuzziness and Soft Computing, vol 141. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-39930-8\_8.

[3] Imran, A., Salhi, S., Wassan, N. A. A variable neighborhood-based heuristic for the heterogeneous flfleet vehicle routing problem. European Journal of Oper-ational Research, 2009, 197 (2), 509–518.

[4] Goldberg, D.E., & Holland, J.H. Genetic Algorithms and Machine Learning. Machine Learning, 1998, 3, 95-99.

[5] 1 X D ,2 ,1 X R , et al. An Improved Tabu Search for the Split Delivery VRP[C]//IEEE Beijing Section,China.Proceedings of 2008 IEEE International Symposium on IT in Medicine and Education(ITME 2008).IEEE Press,2008:4.

[6] Solomon MM. Algorithms for the vehicle routing problem with time window constraints [J]. Operations Research, 1987, 35 (2): 254-265.

[7] Bekta T, Laporte G. The pollution routing problem [J]. Transportation Research Part B: Methodological, 2011, 45: 1232-1250.

[8] 罗梓瑄，杨杰庆，刘学文. 基于 NSGA-II 的考虑客户满意度的多目标车辆路径问题研究[J]. 重庆师范大学学报：自然科学版，2020，37(6)：13-17.

[9] 徐慧英，赵建民，张泳，等. 改进 NSGA-II 算法在车辆路径多目标优化问题中的应用[J]. 计算机工程与科学，2010，32(10)：117-121.

[10] Larsen J. Parallelization of the vehicle routing problem with time windows[M]. Lyngby, Denmark: Institute of Mathematical Modelling, Technical University of Denmark, 1999.

[11] Letchford, A.N., Eglese, R.W., Lysgaard, J.: Multistars, partial multistars and the capacitated vehicle routing problem. Math. Program. 94, 21–40 (2002)

[12] Golden, B. L., Wasil, E. A., Kelly, J. P., & Chao, I.-M. (1998). The Impact of Metaheuristics on Solving the Vehicle Routing Problem: Algorithms, Problem Sets, and Computational Results. In Fleet Management and Logistics (pp. 33–56).

[13] Chris Groër, Bruce Golden, Edward Wasil, (2008) The Consistent Vehicle Routing Problem. Manufacturing&Service Operations Management11 (4):630-643.https://doi.org/10.1287/msom.1080.0243.

[14] Faulin, J. (2003). Toth, Paolo, Daniele Vigo, eds. 2002. The Vehicle Routing Problem. Interfaces, 33(6), 115+. https://link.gale.com/apps/doc/A112799016/AONE?u=anon~928d331a&sid=googleScholar&xid=362a577a.

[15] Mester D, Bräysy O. Active-guided evolution strategies for large-scale capacitated vehicle routing problems[J]. Computers & operations research, 2007, 34(10): 2964-2975.

[16] Imran, A., Salhi, S., & Wassan, N. A. (2009). A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. \*European Journal of Operational Research\*, 197(2), 509-518. DOI: 10.1016/j.ejor.2008.07.022

[17] 夏学文, 刘经南, 高柯夫, 李元香, 曾辉. 具备反向学习和局部学习能力的粒子群算法[J]. 计算机学报, 2015, 38(07): 1397-1407.

[18] Mester, D., Br¨aysy, O. Active-guided evolution strategies for large-scale capacitated vehicle routing problems. Computers & Operations Research, 2007, 34 (10),2964–2975.

[19] Polacek, M., Hartl, R. F., Doerner, K., Reimann, M. A Variable Neighborhood Search for the Multi Depot Vehicle Routing Problem with Time Windows. Journal of Heuristics, 2004, 10 (6),

[20] Polat, O., Kalayci, C. B., Kulak, O., G¨unther, H.-O. A perturbation based variable neighborhood search heuristic for solving the Vehicle Routing Problem with Simultaneous Pickup and Delivery with Time Limit. European Journal of Operational Research, 2015, 242 (2), 369–382.

[21] 王征, 张俊, 王旭坪. 多车场带时间窗车辆路径问题的变邻域搜索算法[J]. 中国管理科学, 2011, 19(2): 99-109.

[22] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. 2017. Learning combinatorial optimization algorithms over graphs. In Advances in Neural Information Processing 6348–6358.

[23] Chaitanya K Joshi, Thomas Laurent, and Xavier Bresson. An efficient graph convolutional network technique for the travelling salesman problem. 2019, arXiv preprint arXiv:1906.01227.

[24] Alex Nowak, Soledad Villar, Afonso S Bandeira, and Joan Bruna. A note on learning algorithms for quadratic assignment with graph neural networks.2017, Stat 1050 , 22.

[25] 李进, 张江华. 基于碳排放与速度优化的带时间窗车辆路径问题[J]. 系统工程理论与实践, 2014, 34(12): 3063-3072.

[26] 南丽君,陈彦如,张宗成.改进的自适应大规模邻域搜索算法求解动态需求的混合车辆路径 问题[J].计算机应用研究,2021,38(10):2926-2934.

[27] Michel Deudon, Pierre Cournut, Alexandre Lacoste, Yossiri Adulyasak, and LouisMartin Rousseau. Learning heuristics for the tsp by policy gradient. In International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Springer,2018, 170–181.

[28] Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning.2016, arXiv preprint arXiv:1611.09940 .

[29] Z. H. Zhan, J. Zhang, Y. Li, et al. Adaptive particle swarm optimization[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2009, 39(6): 1362-1381.

[30] 李英, 张鹏威, 吴一帆. 电动汽车/传统汽车混合车队车辆配置及路径优化模型[J]. 系统管理学报, 2020, 29(03): 522-531.

[31] Clerc, M. (2004). Discrete Particle Swarm Optimization, illustrated by the Traveling Salesman Problem. \*DOI: 10.1007/978-3-540-39930-8\_8\*.

[32] X. B. Gan, L. J. Liu, J. S. Chen, et al. Comprehensive learning PSO for solving environment heterogeneous fixed fleet VRP with time windows[C]. International Conference on Swarm Intelligence. Springer, Cham, 2016: 424-432.

[33] M. Okulewicz, J. Mandziuk. The impact of particular components of the PSO-based algorithm solving the Dynamic Vehicle Routing Problem[J]. Applied soft computing, 2017, 58: 586-604.

[34] Islam, M. A., Gajpal, Y., & ElMekkawy, T. Y. (2021). Hybrid particle swarm optimization algorithm for solving the clustered vehicle routing problem. \*Applied Soft Computing\*, 110, 107655.