****

**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班 |

本科生院制

2024年6月

基于粒子群算法的车辆路径规划

摘要

近年来，我国物流运行保持回升态势，随着实体经济的复苏，物流需求快速增长，物流供应服务体系和供应链进一步完善，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。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。

为了解决上述问题，本研究将实际情况抽象为多目标优化车辆路径问题（VRP），以总成本最低和满意度最高为优化目标，采用改进的粒子群算法（PSO）求解。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-难问题，使用精确求解器很难得到大规模问题的最优解，因此本研究采用了改进的粒子群算法。本研究采用混沌映射法对粒子群进行初始化，以提高粒子群的多样性；采用莱维飞行算法提高粒子的探索和学习能力；采用 k-opt方法对历史最优解进行优化，以提高解的质量。

本研究使用高德的城市拥堵数据、Homberger车辆数据集来验证结果的有效性，并与自适应粒子群算法进行比较。

**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。本课题拟构建配送总成本最低、客户平均满意度最高的优化模型。

车辆路线问题（VRP）最早是由Dantzig和Ramser[1]于1959年首次提出，它是指一定数量的客户，各自有不同数量的货物需求，配送中心向客户提供货物，由一个车队负责分送货物，组织适当的行车路线，目标是使得客户的需求得到满足，并能在一定的约束下，达到诸如路程最短、成本最小、耗费时间最少等目的。由定义可知，著名的旅行商问题（TSP）是VRP问题的一个特例，由于Gaery已证明TSP问题是NP难题，因此，VRP也属于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：

存在一个配送中心D，有M辆车，车辆容积为Q，有N位顾客，每位顾客有其需求量D。车辆从场站出发对客户进行配送服务最后返回场站，要求所有顾客都被配送，每位顾客一次配送完成，且不能违反车辆容量的限制，目的是所有车辆路线的总距离最小。车辆路线的实际问题包括配送中心配送、公共汽车路线制定、信件和报纸投递、航空和铁路时间表安排、工业废品收集等。

在基本车辆路线问题（VRP）的基础上，车辆路线问题在学术研究和实际应用上产生了许多不同的延伸和变化型态，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、多车种车辆路线问题、车辆多次使用的车辆路线问题、考虑收集的车辆路线问题、随机需求车辆路线问题等。

物流是供应链活动的一部分，是为了满足客户的需要，以最低的成本，通过运输、保管、配送等方式，实现原材料、半成品、成品、服务消费以及相关信息由商品的产地到商品的消费地所进行的计划、实施和管理的全过程。物流以仓储为中心，促进生产与市场保持同步。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。

当前物流行业运输成本居高不下，其中运输成本占据大头。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%。21世纪以来,随着经济全球化格局的形成,国内物流行业进入了快速发展通道，物流需求规模持续保持较高增幅，物流业增加值平稳增长。国家统计局数据显示，2013年国内社会物流总额已经达到197.8万亿元，同比增长9.3%,但物流成本过高成为了制约物流业持续、健康发展的重要因素。我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年我国社会物流总费用达到10.2万亿元，占GDP的比重为18%,美国这一比重仅为8.5%,发达国家平均比重为9%,发展中国家平均比重为6%。在居高不下的运输成本构成中，燃油费占比46%,公路、桥梁收费占比25%,两者占比合计达到7成多，其余人工成本、车辆保险损耗占比不到3成。

国家邮政局发文公布 2023 年第三季度快递服务满意度调查和时限准时率测试结果。2023 年第三季度用户快递服务公众满意度得分为 82.9 分，同比上升 0.7 分。测试发现，2023 年第三季度快递服务全程时限为 54.24 小时，同比缩短 2.50 小时。72 小时准时率为 83.75%，同比提升 3.36 个百分点。

从物流运输成本方面来看，数据显示了我国物流成本高企的现状。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。

客户满意度也是物流行业需要重视的重要指标。尽管快递服务的公众满意度得分有所提升，但仍有优化的空间。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。

#todo 把这里的详细介绍放到后面去，平衡文章结构

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。启发式算法则比较多样，如遗传算法，模拟退火方法，禁忌搜索，蚁群算法，变领域搜索等。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。该方法通过将全部可行解空间反复分割为越来越小的子集（分支），并计算每个子集内的解集的目标下界（对于最小值问题）来进行搜索与迭代。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，所以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解，所以精确算法只能在小规模实例中求解，而大规模实例一般都采用启发式算法求解。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的基础上添加了时间窗约束，也就是CVRPTW问题，并且给出了一组算例，被称为Solomon算例，直到今天还在被广泛使用。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。2007年，Bräysy[18]提出了改进版变邻域搜索算法。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。

综上所述，国内外对于车辆路径问题的分类、建模、求解方法等方面做出了各种各样的探索，研究成果十分丰富。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James Kennedy和Russell Eberhart提出，用于解决优化问题。

粒子群算法模拟了鸟群或鱼群等生物个体在搜索过程中的群体行为。在PSO算法中，解空间中的每个潜在解被称为粒子，这些粒子沿着解空间中的搜索方向移动，并根据其个体历史最优解和群体历史最优解来调整移动的方向和速度。

PSO算法的基本思想是通过模拟群体的协作和信息共享来搜索解空间中的最优解。每个粒子都有一个位置向量和一个速度向量，它们通过迭代更新来搜索解空间。粒子的移动受到个体历史最优解和全局历史最优解的影响，以及一定程度的随机性。

粒子群算法的理论研究是对该算法本身的探索和改进。Shi等人将惯性权重加入了粒子群算法的速度更新公式中，并通过大量实验测试了惯性权重对于粒子群算法寻优能力的影响。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优过程中的策略，对于粒子跳出局部最优以及扩大搜索范围具有重要作用。Chaturvedi等人通过对学习因子c1和c2的研究，得出了学习因子的值在[0,2]时，算法能够取得较好的结果。Jin等人提出了一种基于健康度的粒子群算法（HPSO），该算法通过对每一个粒子计算健康度来区分正常粒子和异常粒子，对于每一代中的异常粒子，通过对其个体最优值进行变异操作，提高其健康度。实验结果也表明HPSO具有较好的搜索能力。

因为PSO有容易陷入局部最优解的缺点，一些研究者也着力研究如何提升PSO的全局搜索能力。综合学习粒子群算法（CLPSO）由Liang等人提出，其特点在于更新单个粒子的飞行速度时，综合了整个种群粒子的历史最优信息。由此，该算法能够维持良好的种群多样性，有效地避免了算法过早陷入局部最优解的问题。Zhan[29]等人提出了自适应粒子群优化算法（APSO），其特点在于具备快速收敛的全局搜索能力。该算法根据种群的空间分布信息和粒子的优劣实时计算并确定粒子群的进化状态，并根据种群的不同状态动态调整算法中的各种参数。这样设计能够使得算法具有较好的搜索性能和收敛速度。此外，APSO还引入了精英学习策略，有助于算法跳出局部最优解。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。夏学文等人提出了一种具备反向学习和局部学习能力的粒子群优化算法。该算法在每次迭代中记录下最差粒子的位置以及每个个体的历史最差位置。当算法陷入停滞时，部分粒子采用反向学习策略，向这些较差的信息学习，以使粒子摆脱局部最优。同时，利用种群中较优粒子的位置引导粒子在局部进行精细搜索，有效地提高了算法的局部搜索能力。

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

VRP问题作为NPH问题，使用精确方法求解困难，因此通常使用PSO等启发式算法求解此类问题。Gan[32]等人将碳排放因子引入到车辆路径问题中，提出了一种新的 VRP 模 型并使用 CLPSO 对其进行求解，并将结果与 GA 求解的结果进行对比。实验表明 CLPSO 算法有着较好的求解效果。Okulewicz[33]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 问题的求解中，并在标准测试函数上与其他最新提出的改进 PSO 算法进行比较，结果表明文章所提出的算法得到的结果均值和 收敛速度均优于其他算法。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在解决VRP问题上具有很大的潜力。

1.4 研究内容和方法

1.4.1 研究内容

本研究将实际车辆调度问题，抽象为VRP模型，并使用PSO求解。

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。

针对上述情况，本文使用了以下方法改进算法，并通过仿真实验验证算法的有效性。具体的研究内容如下：

（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，提高种群的多样性，使用莱维飞行（Levy Flight）增强搜索能力，并在最后使用kopt算法提升解的质量。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

本文的研究方法主要包括文献分析和仿真实验，具体使用的研究方法如下：

1. 文献分析法：

通过阅读相关文献，调研业界现状的方式，了解目前VRP问题和PSO算法的研究现状，并对VRP问题和PSO算法进行原理、特点、改进方向等方面的总结分析。

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

本文一共分为五个章节。第一章为绪论，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。第二章为主要研究内容和相关技术方案，主要介绍了本文运用到的相关算法和技术方案。第三章是设计方案与实验细节，主要介绍了算法实现、参数调节的内容。第四章为实验结果与分析，主要讲述了算法的结果和对结果的分析。第五章为总结与展望，总结全文并提出了对于未来的期望。

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（Vehicle Routing Problem，VRP），是指在给定一组客户需求点和一组具有容量限制的车辆的情况下，确定每辆车的路线，使得满足所有客户需求的同时，最小化总行驶距离或总成本。VRP是一个经典的组合优化问题，具有重要的理论和实际应用价值，在物流、运输、配送等领域广泛应用。

VRP问题通常是一个NP-hard问题，因为随着客户数量的增加，搜索空间呈指数增长，难以通过穷举法求解。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问题在现实生活中有着广泛的应用，例如物流配送、城市交通规划、公共交通优化等领域。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. 车辆：每辆车辆只访问客户点一次，且每个客户点仅由一辆车完成服务。
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（HVRP），即车辆不再是同质车辆，存在不同类型的车辆，如不同的容量限制，车辆速度和运行成本。

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。其中配送成本主要来源于车辆行驶耗费的油费和驾驶员的工资，由于驾驶员工资一般由底薪和绩效构成，所以综合考虑可以使用行驶距离作为配送成本最低的优化函数。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. 若车辆未在指定时间段到达客户点，则客户满意度随与软时间窗差距的增大而下降。
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：

#todo：公式

（2-1）

从方程（2-1）可以看出，设TC=，则方程（2-1）被转化为

（2-2）

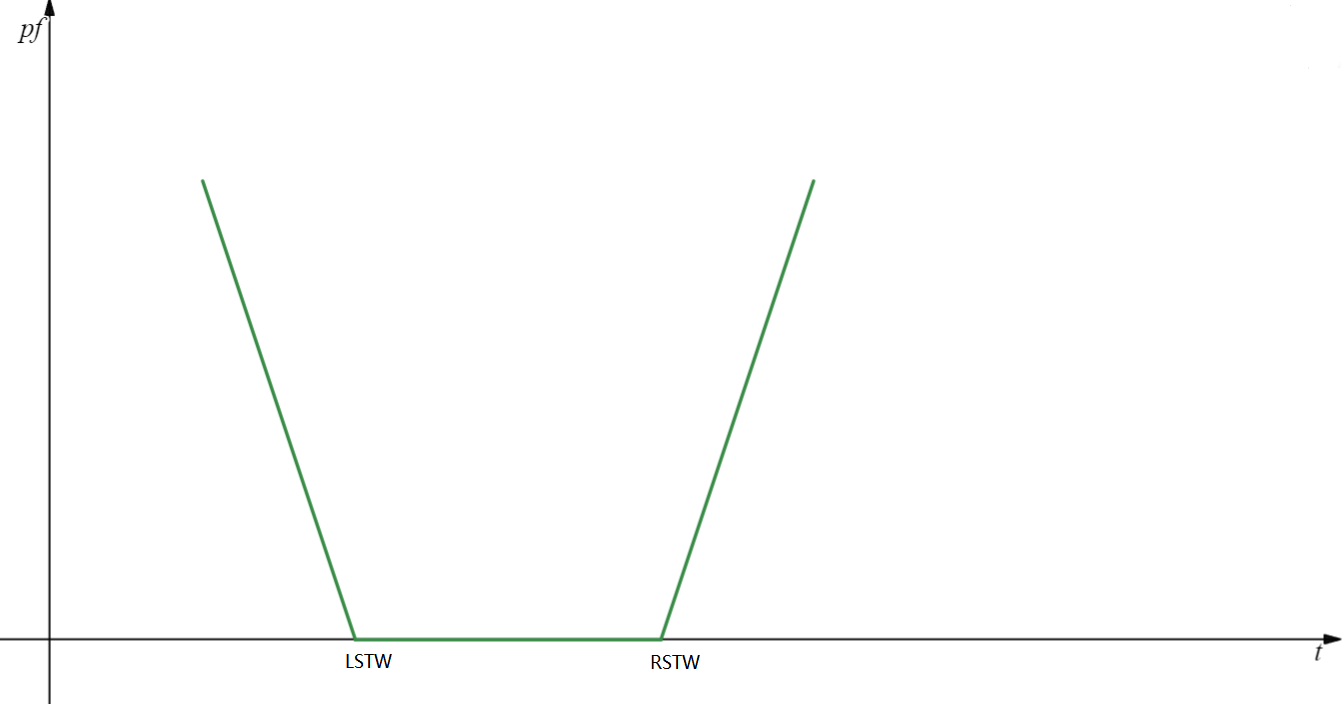
则t2可以通过查表获取，因为TC显然为非递减函数，所以查表时间复杂度为O（log n），在分钟精度下的交通拥堵情况，可以认为TC的反函数ITC以常数时间运行。

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。部分行业晚到的损失大于早到的损失，如急速配送业务。

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。

下面给出本文研究目标的形式化的描述。

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：

#todo：公式

（2-4）

（2-5）

约束：

（2-6）

（2-7）

（2-8）

（2-9）

（2-10）

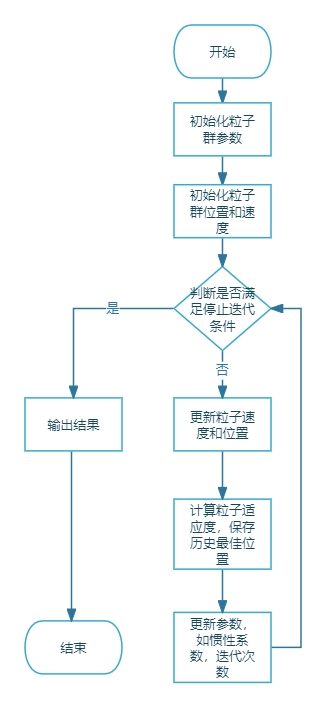
其中公式（2-4）表示最小化车辆行驶距离，公式（2-5）表示最小化用户的不满意度，pf（k）表示计算车辆k服务的客户点的罚函数之和，公式（2-6）表示每个客户点都有且仅有一辆车服务，公式（2-7）表示所有车辆从起点出发，公式（2-8）表示所有车辆回到终点，公式（2-9）表示车辆载货量不超过容积，公式（2-10）为决策变量，表示车辆k从i点前往j点。

2.2 粒子群算法

粒子群算法（PSO）是由James Kennedy和Russell Eberhart观察鸟类群体觅食建立的仿生算法。粒子群算法是一种复杂适应系统（Complex Adaptive System，CAS），复杂适应系统理论认为系统演化的动力本质上来源于系统内部，微观主体的相互作用生成宏观的复杂性现象，其研究思路着眼于系统内在要素的相互作用。

粒子群算法基本原理为生成若干粒子，称为一个种群，每个粒子都有一个位置向量和速度向量，位置表示一个可行解，速度表示位置的变化率，并且粒子能感知当前解的适应度，粒子能记录自己的历史最佳位置和感知到种群历史最佳位置，并且根据学习率、自己和种群粒子最佳位置改变速度向量，靠近最佳位置，以此完成搜索。

粒子群算法的流程可以表示为，首先初始化粒子群参数，如粒子群数量、迭代最大次数、惯性权重，然后初始化粒子，包括粒子的位置和速度，然后计算适应度，判断是否满足停止迭代的条件，如果满足则输出结果，结束流程，如果不满足则更新粒子，计算适应度，保存种群和粒子的历史最佳位置，更新参数，然后再次判断是否能够停止迭代。具体流程图如下。



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

每一个粒子位置由向量，组成，，都是n维向量，每一个粒子速度也是两个n维的向量构成，记为，。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。

#todo：考虑是否需要添加伪代码

由于本研究的目标函数有多个，所以我们采用多目标优化的粒子群算法。

常见的多目标优化决策方法有权重聚合方法、基于惩罚的边界交叉方法、帕累托前沿方法等等。

其中权重聚合方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。

2.3 改进粒子群的方案

虽然PSO思想简单，易于实现，不需要优化函数有良好的性质，收敛速度快，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

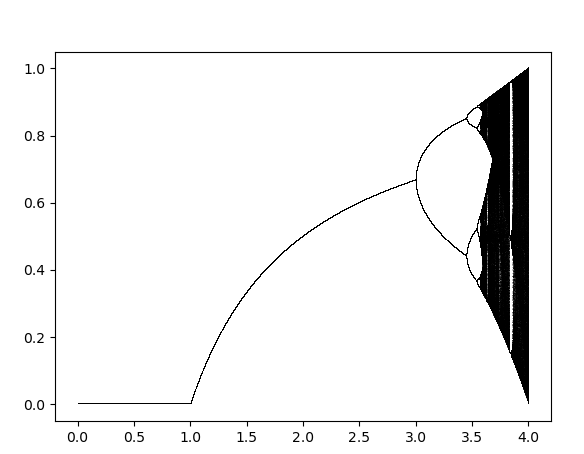
混沌映射算法常用于生成随机数，在优化领域，经过混沌映射生成的初始种群通常由更好的种群多样性。

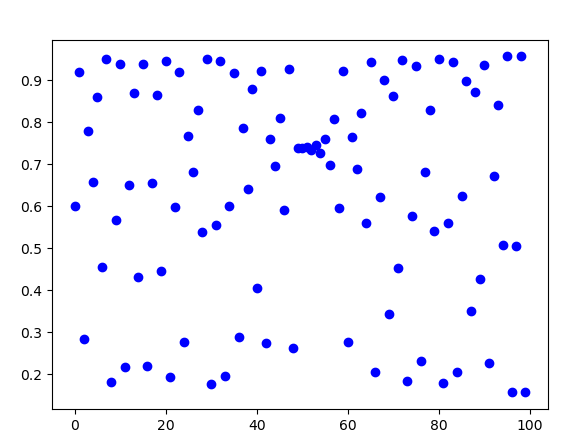
虽然混沌映射算法能提高初始种群多样性，但是经过实验，对于最终结果无明显提升，仅在初期加快了算法收敛的速率。

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∈(3.57,4]时，系统处于混沌状态,具体生成公式和样例如下。

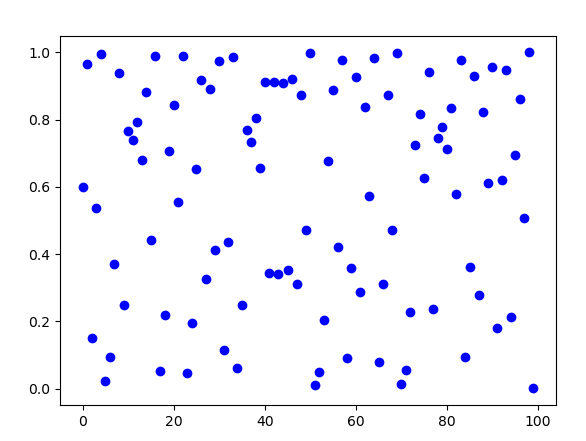
（2-10）





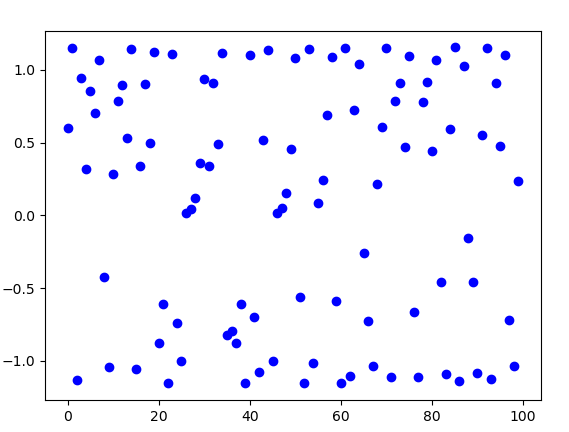
1. logisticsSine映射：与Logistic映射类似，但是添加了一项与正弦函数相关的项，具体生成公式和样例如下。

（2-11）



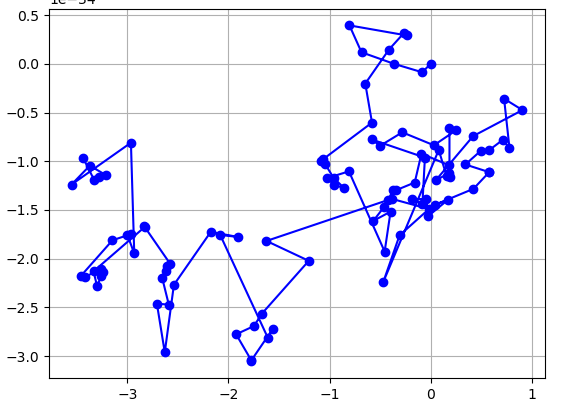
1. Cubic映射：与逻辑混沌映射相比，Cubic映射具有更好的混沌遍历能力。它的优化速度快、精度高，具体生成公式和样例如下。

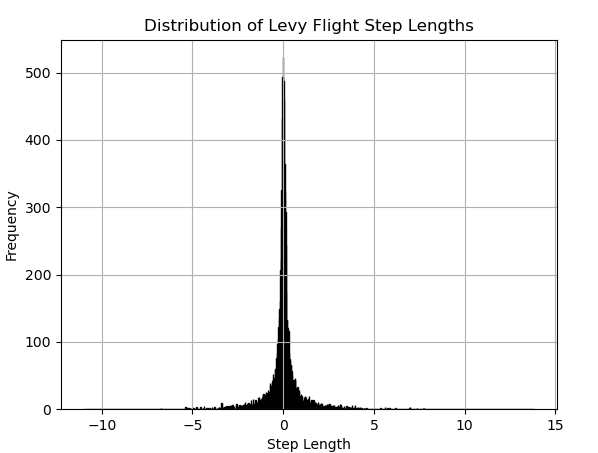
（2-12）



2.3.2 Levy飞行

莱维飞行（Levy flight）以法国数学家保罗·莱维命名，指的是步长的概率分布为重尾分布的随机行走，也就是说在随机行走的过程中有相对较高的概率出现大跨步。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）。因此，很多随机行走过程都可以用莱维飞行来描述。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复杂，一般使用Mantegna方法生成，具体公式如下：

（2-12）

（2-13）

（2-14）

（2-15）

其中β为控制随机变量分布的参数。

本研究使用莱维飞行在粒子学习个体历史最优位置和群体历史最优位置时增强学习的速率。具体改进公式如下：

ition（2-16）

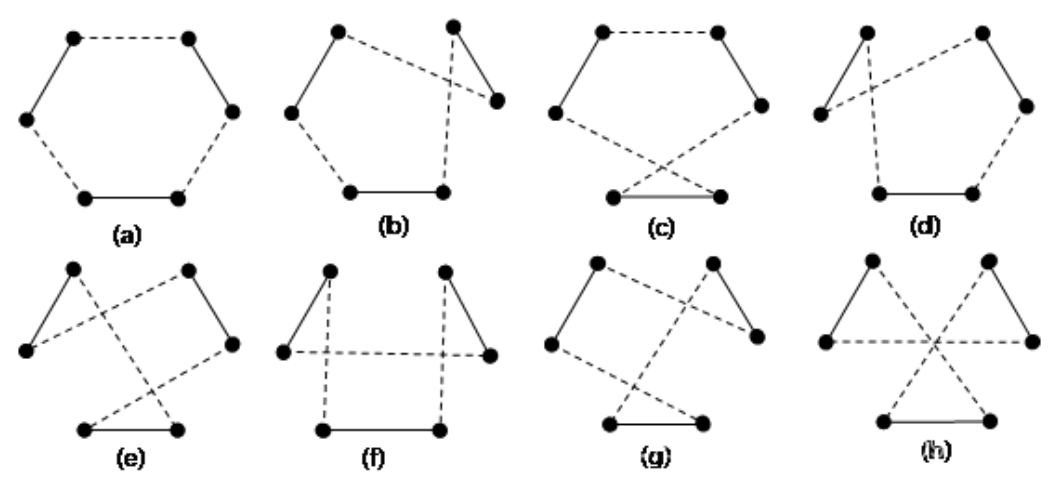
其中c为c1或c2，s为莱维飞行生成的随机变量的值，为个体历史最优位置或群体历史最优位置与当前位置的差值。

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kopt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。

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。对于一个新的路径，如果新路径优于原先的路径，则保留新的路径，删除原先的路径。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。

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算法与原始的PSO算法、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

进化状态评估采用欧氏距离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为粒子群的种群数量，D为粒子的维度大小。

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）

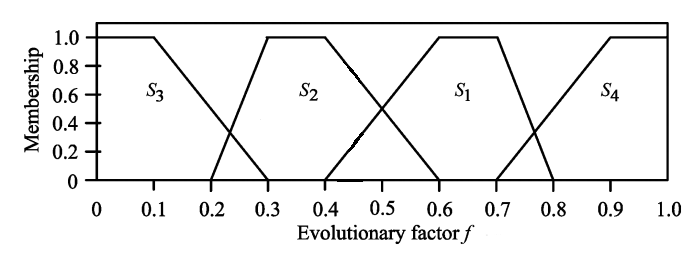
根据f与隶属函数判断当前系统处于所处的阶段。根据参考文献，系统一共有4个状态，分别是S1（探索），S2（发现），S3（收敛），S4（跳出），对应的隶属函数和分布图如下：

（2-19）

（2-20）

（2-21）

（2-22）



可以看到部分区域存在重叠，重叠部分我们根据隶属度大小来划分状态，即隶属度大的为被选择的状态。

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）

1. 设计方案和实验细节

3.1 实验环境

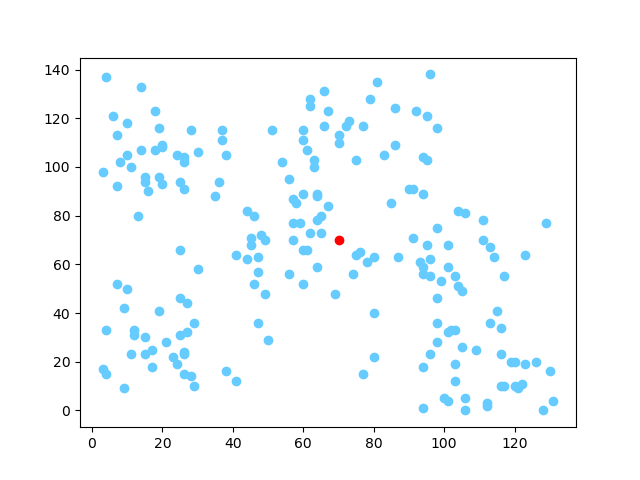
实验环境的系统为Windows 10专业版19045.4291，处理器为AMD Ryzen 5 4600H with Radeon Graphics（3.00 GHz），机带RAM大小为16.0 GB (15.4 GB 可用)。Python环境为Python3.9.12。

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=1000次，粒子群大小N=300，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。下面为一个实例的客户点分布展示。



其中红色的点为仓库，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

3.3.3 算法流程图

