

## 摘要

随着经济的快速发展和城市化程度的提高,在人们生活和生产过程中物流活动急剧增加。人们对高效合理的物流配送方案的需求越来越迫切,同时,物流活动带来的环境问题也随着国家对绿色经济的提倡得到越来越多学者的关注。车辆路径优化问题作为物流活动的根本问题,是通过对车辆行驶路线进行高效合理的规划,以达到运输成本最小化的一类优化问题,其相关理论和算法对于现实生活有着不可缺少的价值。在物流配送过程中,交通拥堵时常发生,如何在复杂情况下及时满足客户需求,最小化运输成本并降低对环境的污染成为广泛关注且具有现实意义的课题。

本文的主要研究内容和创新点如下:

(1)时变环境下的带时间窗绿色多目标车辆路径优化问题模型的构建。现有的带时间窗约束的车辆路径优化问题模型中对现实条件的考虑较少,往往只考虑了时间窗约束,且现有模型考虑的求解目标单一,不能很好的反映实际配送过程对路径规划的要求。对于以上不足,本文提出一种时变环境下的带时间窗绿色多目标车辆路径优化问题(Time Dependent Green Vehicle Route Problem with Time Window, TDGVRPTW)模型,该模型同时考虑三个目标,包括总运输成本、总行驶时间以及车辆数。并对容量、时间窗和道路拥堵情况等约束以及碳排放量等成本进行考虑,相比以往模型更加符合实际情况。

(2)提出了一种改进多目标差分进化算法求解 TDGVRPTW。随着车辆路径优化问题求解目标数量的增加,算法求得优质解的困难明显增加,为了求得更好的解集,提出一种改进差分进化算法用于求解 TDGVRPTW。算法的改进主要体现在初始化阶段和变异算子以及选择策略:在初始化阶段,提出反向 Tent 混沌初始化生成分布更加均匀的初始化种群,提高算法的搜索范围;在进化过程中,设计了一种混合变异策略以及修补算子,混合变异策略融合两种流行 DE 算子,修补算子对越界个体进行修补,防止后代种群越界后聚集在解空间边界上;在选择阶段,设计一种自适应选择策略保留潜在解。提出的改进策略较好的平衡了种群的多样性和收敛性。通过理论分析以及在 Solomon 基准数据集以及真实 MOVRPTW 实例进行对比实验验证提出的改进算法的有效性。

(3)提出了一种基于协同进化框架的动态约束多目标优化算法求解 TDGVRPTW。VRP 问题具有复杂的约束条件,而现有的 VRP 求解算法中,往往根据约束条件做出严格限制以满足问题要求,未能充分利用约束条件信息对问题求解的作用。因此基于多任务优化中简单任务辅助复杂任务求解的思想,设计了一种合作协同约束多目标进化算法用于求解 TDGVRPTW。该算法将完整的 TDGVRPTW 作为复杂任务,在求解复杂任务时,选择过程引入同时考虑个体的分布和收敛信息的移位拥挤距离计算,较好的平衡种群的收敛性和多样性。同时设计了一种动态约束选择策略用于简单任务的实现,简单任务考虑部分约束条件对种群的影响。任务之间通过子代种群进行信息交互。在约束多目标测试函数集、Solomon 基准数据集以及真实 MOVRPTW 实例进行对比实验,实验结果表明基于协同进化框架的约束多目标优化算法具有较好的综合性能。

关键词: 车辆路径优化; 碳排放; 时间窗; 多目标优化; 协同进化; 差分进化

## Abstract

With the rapid development of the economy and urbanization, logistics activities dramatically increased in people's daily life and production processes. The need for efficient and reasonable logistics delivery solutions was becoming increasingly urgent, while environmental problems caused by logistics activities also received more attention from scholars, especially with the country's promotion of a green economy. Vehicle routing optimization problem was fundamental to logistics activities, aiming to minimize transportation costs by planning the most efficient and reasonable vehicle driving routes. Its related theory and algorithms have invaluable value for real-life applications. However, in the process of logistics distribution, traffic congestion often occurs. How to meet customer needs in a complex situation, minimize transportation costs and reduce environmental pollution has become a subject of widespread concern and practical significance.

The main research contents and innovative points of this paper include:

(1) Constructing a Time Dependent Green Vehicle Route Problem with Time Window (TDGVRPTW) model in a time-dependent environment. Existing models for vehicle routing optimization problems with time window constraints considered few realistic conditions and often only focused on time window constraints. Moreover, existing models considered single optimization objectives that could not reflect the requirements of practical distribution processes for path planning well. To address these issues, this paper proposed a TDGVRPTW model that considers three objectives simultaneously, including total transportation costs, total travel time, and the number of vehicles. The model also considered constraints such as capacity, time windows, road congestion, and cost factors such as carbon emissions, making it more consistent with actual situations than previous models.

(2) An improved multi-objective differential evolution algorithm was proposed to solve the TDGVRPTW problem. As the number of optimization objectives for vehicle routing optimization problems increases, it becomes increasingly challenging to obtain high-quality solutions through algorithms. To obtain better solution sets, this paper proposed an improved differential evolution algorithm to solve the TDGVRPTW. The improvements mainly focused on initialization, mutation operators, and selection strategies. In the initialization stage, reverse Tent chaotic initialization was proposed to generate more uniformly distributed initial populations and expand the search scope of the algorithm. During the evolutionary process, a hybrid mutation strategy and repair operator were designed. The hybrid mutation strategy integrated two popular differential evolution operators, while the repair operator repaired out-of-range individuals to prevent offspring populations from clustering on the boundary of the solution space. In the selection stage, an adaptive selection strategy was designed to retain potential solutions. This strategy balanced population diversity and convergence well. The effectiveness of the proposed improvement algorithm was verified through theoretical analysis and comparative experiments on the Solomon benchmark dataset and real MOVRPTW instances.

(3) A dynamic constraint multi-objective optimization algorithm based on the cooperative

co-evolution framework was proposed to solve the TDGVRPTW. VRP problems have complex constraints, and existing VRP algorithms often impose strict restrictions based on constraints to meet problem requirements, failing to fully utilize the role of constraint information in solving problems. Therefore, based on the idea of simple tasks assisting complex task solving in multi-task optimization, a collaborative constraint multi-objective evolutionary algorithm was designed to solve the TDGVRPTW. The algorithm treated the complete TDGVRPTW as a complex task and introduced a shift crowding distance calculation that simultaneously considered the distribution and convergence information of individuals in the selection process, which balanced the convergence and diversity of the population well. Meanwhile, a dynamic constraint selection strategy was designed to implement simple tasks, considering the impact of partial constraint conditions on the population. The two populations exchanged information through offspring populations. Comparative experiments were conducted on constraint multi-objective test function sets, Solomon benchmark datasets, and real MOVRPTW instances, and the experimental results showed that the constraint multi-objective optimization algorithm based on the cooperative co-evolution framework had good comprehensive performance.

**Keywords:** Vehicle Routing Optimization; Carbon Emission; Time Window; Multi-objective Optimization; Coevolution; Differential Evolution

# 目 录

第一章 绪论.....	1
1.1 课题背景及研究意义.....	1
1.2 国内外研究现状及发展动态 .....	2
1.2.1 车辆路径问题研究现状.....	2
1.2.2 带时间窗的车辆路径问题研究现状.....	3
1.2.3 绿色车辆路径问题研究现状.....	4
1.3 本文研究内容.....	5
1.4 论文结构安排.....	6
第二章 相关理论基础 .....	7
2.1 车辆路径优化相关理论.....	7
2.1.1 时间窗类型.....	7
2.1.2 时间依赖函数.....	7
2.1.3 碳排放模型.....	8
2.1.4 相关数据集.....	9
2.2 多目标优化相关理论.....	11
2.2.1 多目标优化问题定义.....	11
2.2.2 差分进化算法概述.....	11
2.2.3 NSGA II 算法概述.....	13
2.2.4 多目标优化性能评价指标.....	14
2.3 本章小结.....	14
第三章 带时间窗约束的时变绿色多目标车辆路径问题模型构建 ....	15
3.1 带时间窗约束的时变绿色多目标车辆路径问题 .....	15
3.2 问题描述.....	15
3.3 模型构建.....	16
3.3.1 模型假设.....	16
3.3.2 符号定义.....	16
3.3.3 行驶时间计算.....	17
3.3.4 TDGVRPTW 模型.....	18
3.4 本章小结.....	19
第四章 基于改进差分进化算法的车辆路径优化 .....	21
4.1 基于改进初始化和变异选择策略的差分进化算法 .....	21
4.1.1 反向混沌初始化.....	21
4.1.2 变异策略及修补算子.....	22
4.1.3 自适应选择策略.....	24

4.1.4 IMSDE 算法描述 .....	25
4.1.5 IMSDE 算法复杂度分析 .....	26
4.1.6 IMSDE 算法性能分析 .....	26
4.2 基于 IMSDE 算法的车辆路径优化 .....	34
4.2.1 算法应用流程 .....	34
4.2.2 实验设置 .....	35
4.2.3 Solomon 数据集结果分析 .....	36
4.2.4 真实数据集结果分析 .....	39
4.3 本章小结 .....	40
第五章 基于合作协同进化算法的车辆路径优化 .....	41
5.1 基于合作协同进化框架的约束多目标优化算法 .....	41
5.1.1 CDSMO 算法思想与框架 .....	41
5.1.2 动态约束选择策略 .....	41
5.1.3 移位拥挤距离 .....	43
5.1.4 CDSMO 算法描述 .....	45
5.1.5 CDSMO 算法复杂度分析 .....	46
5.1.6 CDSMO 算法性能分析 .....	46
5.2 基于 CDSMO 算法的车辆路径优化 .....	54
5.2.1 算法应用流程 .....	54
5.2.2 实验设置 .....	54
5.2.3 Solomon 数据集结果分析 .....	55
5.2.4 真实数据集结果分析 .....	57
5.3 本章小结 .....	58
第六章 主要结论与展望 .....	59
6.1 主要结论 .....	59
6.2 展望 .....	60
参考文献 .....	61

## 第一章 绪论

### 1.1 课题背景及研究意义

随着国内物流基础设施布局的完善,电子商务的发展,物流业已经成为国民经济发展中关键环节之一,它连接社会经济的各个部分并使之成为一个有机整体,在国民经济和社会发展中发挥着重要作用。图 1-1 显示了对国家发展改革委公布的近年来的中国社会物流总费用统计情况。可以看出,2022 年我国物流总费用为 17.8 万亿元,较 2021 年增长 6.6%,我国 2015 年至 2022 年总费用不断上升,从 2015 年的 10.8 万亿元攀升到了 2022 年的 17.8 万亿元。物流总费用主要由三部分组成,分别是运输费用、保管费用和管理费用。在 2022 年,运输费用为 9.55 万亿元,占比高达 53.6%,而根据交通运输部数据显示,公路运输量在运输占比高达 75%。由此可见,车辆的运输在整个物流服务过程中占据着非常重要的地位。合理高效的车辆路径规划对物流行业降本增效起着至关重要的作用。

车辆路径问题(Vehicle Routing Problem,VRP)在现实生活中的许多问题中均有所体现<sup>[1]</sup>,例如供应链管理<sup>[2]</sup>,垃圾回收<sup>[3]</sup>,物流调度与运输<sup>[4]</sup>,快递配送<sup>[5]</sup>等。车辆路径问题是指在一定时间内为满足用户需求的多个地点提供服务,对若干车辆进行路线规划,从而使总体费用以及服务车辆的数量最小的问题。高效的路径规划方案可以在满足客户多种多样的要求下,降低车辆的运输成本,提高配送中心的效率以及服务质量,这对国民经济的发展有着极大的促进作用,具有极其重要的意义。

车辆路径问题作为物流配送领域的重要研究课题之一,有着广泛的应用前景,并受到了越来越多的相关领域学者的重视。根据现实问题中不同的要求,相关研究人员将车辆路径问题拓展出以下不同类别:带容量约束的车辆路径问题,带时间窗的车辆路径问题,需求可拆分的车辆路径问题,绿色车辆路径问题,同时送取货的车辆路径问题,车辆异构的车辆路径问题等。为了有效解决各种各样的车辆路径调度问题,学者们提出了多种求解方法,多数方法需要建立数学模型,通过定义不同类型变量、约束函数、目标函数来完成车辆路径的优化。常用的方法主要包括精确搜索算法、启发式搜索算法和元启发式等。

随着客户时间观念的逐渐加强,带时间窗的车辆路径问题(Vehicle Routing Problem with Time Windows,VRPTW)逐渐成为了车辆路径问题研究的一大热点。除此之外,近年来,全球经济发展迅速,能源的消耗也随之大幅增长,导致全球温室气体大量排放,节能减排对于世界来说刻不容缓,成为了世界各地聚焦的热点。我国在 2021 年承诺在 2025 年初步形成绿色低碳循环发展的经济体系,将单位国内生产总值二氧化碳排放比 2020 年下降 19%。根据全球能源署公布数据,交通运输产生的二氧化碳占比高达 24.34%,而公路运输在交通运输中的占比高达 74.4%。因此为了我国战略目标的达成,减少车辆运输过程中造成的二氧化碳排放,对考虑碳排放的车辆路径问题进行研究,有着重大意义。

车辆运输过程中燃油的消耗是二氧化碳排放的主要来源,并且物流企业的成本和效

率也与油耗有着极其紧密的关联。当客户对配送服务时间有着一定的要求时，如何在满足客户需求和最小化碳排放量的情况下，对配送任务做出合理高效的路径规划是研究的重点。随着国家双碳要求的提出，低碳车辆路径规划问题已经渐渐成为物流领域未来发展的主流，对我国节能减排有着极其重要的意义。因此，本文将对复杂交通情况下带时间窗约束的绿色多目标车辆路径问题进行研究。

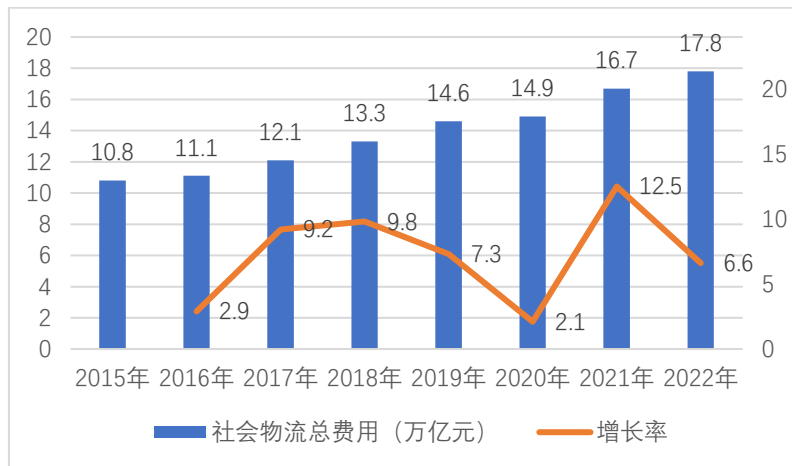


图 1-1 2015-2022 年中国社会物流总费用及其增长率

## 1.2 国内外研究现状及发展动态

### 1.2.1 车辆路径问题研究现状

车辆路径问题由 Dantzig 和 Ramser 于 1959 年<sup>[6]</sup>第一次提出，用于研究亚特兰大炼油企业与各下属加油站之间的油罐车路径优化问题。VRP 由于在现实生活中多个方面均有体现，有着极大的经济价值，一直受到国内外学者的广泛关注。一般意义下的 VRP 可描述为<sup>[7]</sup>：在约束条件下，设计从一个或多个初始点出发，到多个不同位置的城市或客户点的最优送货巡回路径，如图 1-2 所示。

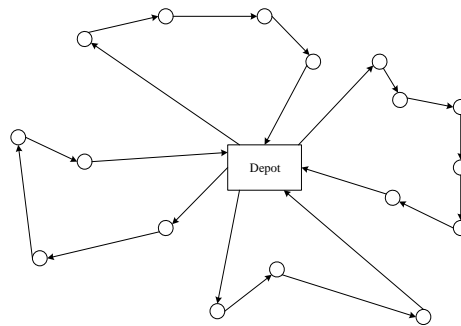


图 1-2 VRP 示意图

在基本 VRP 的基础上，国内外学者在学术研究和现实应用上对 VRP 进行了很多的扩展延伸，已经被扩展和延伸成许多不同的型态，主要包括<sup>[8][9][10][11][12]</sup>：

- (1)带容量约束的车辆路径问题(Capacitated Vehicle Routing Problem, CVRP)
- (2)带时间窗的车辆路径问题(Vehicle Routing Problem with Time Window, VRPTW)
- (3)需求可拆分车辆路径问题(Split Delivery Vehicle Routing Problem, SDVRP)
- (4)动态车辆路径问题(Dynamic Vehicle Routing Problem, DVRP)

(5)随机需求车辆路径问题(Vehicle Routing Problem with Stochastic Demand, VRPSD)

(6)同时送取货的车辆路径问题(Vehicle Routing Problem with Simultaneous Delivery Pickup, VRPSDP)

(7)绿色车辆路径问题(Green Vehicle Routing Problem, GVRP)

为了解决 VRP, 近年来, 国内外学者提出了多种求解方法, 常用的方法主要包括精确搜索算法和启发式搜索算法<sup>[13][14][15]</sup>。其中, 精确搜索算法是针对某一具体问题建立相应的数学模型, 然后利用数学方法进行求解, 一定可以求出问题的最优解, 主要集中在分枝定界法<sup>[16]</sup>、分枝切割法<sup>[17]</sup>、列生成法<sup>[18]</sup>、动态规划法<sup>[19]</sup>等。由于 VRP 是一个 NP 难问题, 使用精确算法在求解时需引入严格的数学方法, 因此在计算时会消耗太多的计算能力和存储空间, 会限制最优车辆路径的准确性, 只能将该方法应用于小规模 VRP 求解中。随着问题规模的逐渐增加, 一些学者提出使用启发式搜索算法来解决 VRP。启发式算法其基本思想是在可接受的范围内给出待解决问题的一个可行解。在 VRP 中主要使用的启发式搜索算法包括粒子群优化算法<sup>[20]</sup>、蚁群算法<sup>[21]</sup>、遗传算法<sup>[22]</sup>、禁忌搜索<sup>[23]</sup>及其改进类型<sup>[24][25]</sup>。相较于精确搜索算法, 启发式搜索算法在处理大规模 VRP 问题时有更好的鲁棒性与可行性。基于现有研究可以看出, 精确搜索算法和智能启发式搜索算法均可用于求解 VRP 及相关问题。精确算法可以找到问题的最优解, 但它高度依赖于问题模型中的解空间、约束的数量和决策变量的数量, 也无法提供不同类型变量、目标和约束的通用解决方案策略。然而, 当问题的规模变得更大时, 精确搜索算法会消耗过多的计算能力和存储空间。通过设计启发式函数, 启发式算法可以在很短的时间内获得搜索问题的最优解。对于 NP-hard 问题, 它也可以在多项式时间内得到更好的解。启发式算法可以进一步提高车辆路径的准确性。

国内的车辆路径优化问题研究开始于上世纪 90 年代, 随着社会的发展和物流业的发展, 该问题得到了越来越多的关注。在国内, 研究者主要集中在 CVRP、VRPTW、GVRP 等问题上。国外的 VRP 研究也有较长的历史, 早在 20 世纪 50 年代, 就有学者开始研究该问题。目前, 国外的研究主要集中在 VRP、VRPTW 等问题上, 同时也涉及到多种变种问题。

### 1.2.2 带时间窗的车辆路径问题研究现状

随着生活节奏的加快, 人们对于时间的要求越来越高。因此带时间窗约束的车辆路径规划问题受到越来越多学者的关注。VRPTW 不仅要在地理空间上选择合理的路径最小化成本, 还要在时间层面上根据客户的服务时间安排有效的客户服务次序。VRPTW 根据时间的不同要求可以划分为硬时间窗<sup>[26]</sup>和软时间窗<sup>[27]</sup>。硬时间窗要求车辆必须在客户给定时间窗内为客户提供服务, 早到需要等待, 迟到就拒收。软时间窗则不一定要在时间窗内开始服务客户, 但是在时间窗外开始服务必须要惩罚, 以惩罚代替等待与拒收是软时间窗和硬时时间窗的最大的区别。

Solomon et al 1987<sup>[28]</sup>首次提出了 VRPTW, 要求提供服务的配送车辆必须在客户给定的时间范围内进行配送, 并给出了该类问题的经典算例。Wang 等人<sup>[29]</sup>提出一种基于多蚂蚁系统的服务时间定制的混合启发式算法用于求解 VRPTW。宾松等人<sup>[30]</sup>设计了一



个改进遗传算法用于求解软时间窗的 VRPTW。曹二保等人<sup>[31]</sup>提出一种混合禁忌搜索和差分进化的算法对 VRPTW 采用次序优化进行求解。Gehring 等人<sup>[32]</sup>和 Alvarenga 等人<sup>[33]</sup>将带时间窗的多目标车辆路径问题(MOVRPTW)分成两阶段进行求解, 第一阶段优化车辆数, 第二阶段优化成本。Tian 等人<sup>[34]</sup>提出一个以 NSGAII 为基础算法的协同进化框架, 种群间采用弱合作的方式进行信息交互用于求解 MOVRPTW。张凯庆等人<sup>[35]</sup>考虑软时间窗约束和车辆速度变化情况, 构建了最大化平均客户满意度、最短配送距离和最小配送成本为目标的优化模型, 并设计了两阶段求解算法对 MOVRPTW 进行求解。Hou 等人<sup>[36]</sup>提出一种面向解评价的多目标差分进化算法, 并提出了一种基于可行解的差分变异策略用于 MOVRPTW 的求解。雷金羨等人<sup>[37]</sup>针对传统蚁群算法容易陷入局部最优的问题增加一个奖励值保证前期迭代搜索的质量进行改进, 将改进的算法应用在 VRPTW。Ghoseiri 等人<sup>[38]</sup>使用目标规划表述多目标 VRPTW, 并设计了一种多目标遗传算法进行求解。Iqbal 等人<sup>[39]</sup>针对人工蜂群激发蜜蜂的觅食动作, 结合混合超启发式算法求解了 VRPSTW。Zhen 等人<sup>[40]</sup>考虑带时间窗和交货时间的多站点多行程的车辆路径问题, 提出了一种混合粒子群优化算法和一种混合遗传算法用于求解该模型。Cheng 等人<sup>[41]</sup>将多任务优化框架与现有的 MOEA 结合提出了一种新的多任务优化算法用于求解 VRPTW, 该算法设计了一种协作机制, 自适应的在知识迁移过程和局部搜索过程之间搜索。Jin 等人<sup>[42]</sup>提出一种结合禁忌搜索和人工免疫算法的混合算法用于求解 VRPTW。Zhang<sup>[43]</sup>等人将快速采样策略全局搜索和基于路径序列差异局部搜索相结合, 设计了一个混合多目标进化算法用于求解 VRPTW。Melián-Batista 等人<sup>[44]</sup>设计了一种基于散点搜索的元启发式算法用于求解以最小化行驶距离和司机工作时间平衡的双目标 VRPTW。

### 1.2.3 绿色车辆路径问题研究现状

近年来, 随着物流运输活动对生态环境的破坏越来越严重, 人们越来越关注交通运输对人类和生态系统的直接或间接的有害影响。根据欧洲环境署的数据显示, 交通运输在温室气体排放总量中的占比高达 20%, 这对交通运输业带来了巨大挑战。Erdoğan 和 Miller-Hooks<sup>[45]</sup>在 2012 年引入了绿色车辆路径问题(Green Vehicle Routing Problem, GVRP)作为经典 VRP 的扩展。在之后的研究中, GVRP 主要的探讨方向是车辆的速度、载重以及车辆类型对运输过程中碳排放量的影响。

Bektas 和 Laporte<sup>[46]</sup>将行程速度作为决策变量构建了污染路径问题, 根据车辆的速度和载重计算油耗和碳排放量。Kunnappadeelert 等人<sup>[47]</sup>将差分进化算法应用于求解带时间窗的绿色车辆路径问题, 根据行驶距离对碳排放量进行计算, 未考虑速度和载重等其他因素。Demir 等人<sup>[48]</sup>使用自适应大邻域搜索启发式算法求解了以燃料消耗和行驶时间的加权和作为目标的污染路径问题。Poonthalir 等人<sup>[49]</sup>采用带时变加速度系数的粒子群优化算法对具有变速约束的 GVRP。Niu 等人<sup>[50]</sup>提出一种膜启发多目标算法用于求解具有随机需求的 GVRP。Franceschetti 等人<sup>[51]</sup>将交通拥堵考虑到带硬时间窗的 GVRP 中, 该模型将速度按拥堵时段的固定速度和非拥堵时段的自由速度进行划分。Ubeda 等人<sup>[52]</sup>给出了一个 GVRP 的真实案例, 其中碳排放量的计算考虑了车辆的载重和行驶距离。Huang 等人<sup>[53]</sup>研究了时变环境下的 GVRP, 道路在不同的时段具有不同的速度, 并对现

实交通实例进行分析,验证了考虑不同时变速度的 GVRP 降低了碳排放量和运输成本。唐金环等人<sup>[54]</sup>提出一种混合粒子群算法用于求解时变环境下的 GVRPTW,行驶时间和碳排放量均基于速度变化,速度则根据交通状况进行计算。任腾等人<sup>[55]</sup>将信息素浓度加入蚁群算法,用于求解以碳排放量最小为目标的 VRP。杨眉<sup>[56]</sup>对在时变网络下研究电动冷藏车配送路径优化问题进行研究,并设计了一种节约里程-禁忌搜索混合启发式算法对模型求解。范厚明等人<sup>[57]</sup>研究了考虑油耗成本的多中心车辆路径问题,根据问题特征设计了混沌遗传模拟退火算法对问题进行求解。

从以上国内外研究现状来看,VRP 目前的相关研究较多,现实生活中的各种条件抽象成问题的约束或者目标,因此产生了多种拓展问题。其中对 GVRP 和 VRPTW 的研究随着城市化程度的提高而日益增加。而综合考虑碳排放量、时间窗约束、速度时变以及载重变化等因素的多目标车辆路径问题的相关文献较少。

关于模型,大部分模型的求解目标单一,部分多目标则以加权和方式转换为单目标进行计算,而现实世界中的问题通常涉及相互冲突的目标,这些目标应同时解决。关于模型求解算法,随着目标数量的增加,算法求解的质量明显下降,还有就是 Pareto 前沿所需的非劣解的数量呈指数增加,这些都为求解更优质的路线带来了困难。同时可以发现 VRP 问题具有各种复杂的约束条件,如时间窗约束和容量约束,而在求解 VRP 问题时,很少有学者对约束信息进行关注并利用。

### 1.3 本文研究内容

对于现有车辆路径问题的模型构建以及求解算法中存在的不足,本文对车辆路径问题进行了更贴近现实情况的模型构建,对围绕该模型设计了相关算法进行求解。本文的主要研究内容如下:

(1)时变环境下的带时间窗绿色多目标车辆路径问题模型的构建。

现有的带时间窗约束的车辆路径问题模型中对现实条件的考虑较少,往往只考虑了时间窗约束,且现有模型考虑的求解目标单一,也不能很好的反映实际物流配送过程对路径规划的要求。对于以上不足,本文提出一种在时变环境下的带时间窗绿色多目标车辆路径优化问题(TDGVRPTW),该模型同时考虑了三个目标,包括运输成本(包含行驶成本,车辆固定成本,时间延迟成本以及受距离、速度和载重影响的碳排放成本)、总行驶时间以及车辆数。该模型同时对容量、时间窗以及道路拥堵情况等约束进行考虑,相比以往模型更加符合现实需求。

(2)提出了一种基于改进初始化和变异策略以及选择策略的约束多目标差分进化算法用于求解 TDGVRPTW。

随着车辆路径问题求解目标数量的增加,算法求得优质解的困难明显上升,为了求得更好的解集,提出一种基于改进初始化和变异策略的差分进化算法用于求解 TDGVRPTW。算法的改进主要体现在初始化方法、变异算子以及选择策略:在初始化阶段,引入 Tent 混沌映射和反向学习生成分布更加均匀的初始化种群,提高了算法的搜索范围;然后设计了一种混合变异策略以及修补算子,首先融合两种流行 DE 算子

DE/rand-to-best/1/bin 和 DE/current-to-rand/1 实现多样性和收敛性的平衡, 然后针对产生后代个体越界后聚集在解空间边界上, 设计了一种平均向量修补算子对越界个体进行处理以保持种群解空间的多样性; 选择过程中, 设计自适应选择策略保留部分潜在优质解, 提高了种群的多样性和收敛性。在约束多目标测试函数集 CF、Solomon 基准 VRPTW 数据集以及真实 VRPTW 实例进行对比试验, 实验结果表明改进约束多目标差分进化算法具有一定优势。

(3)提出了一种基于协同进化框架的约束多目标进化算法用于求解 TDGVRPTW。

VRP 问题具有各种复杂的约束条件, 而现有的车辆路径优化问题求解算法中, 往往根据约束条件做出严格限制以满足问题要求, 未能充分利用约束条件信息对问题求解的作用。因此基于多任务优化中简单任务辅助复杂任务求解的思想, 设计了一种合作协同约束多目标进化算法用于求解 TDGVRPTW。该算法将完整的 TDGVRPTW 作为复杂任务, 在求解复杂任务时, 引入了同时考虑个体的分布和收敛信息的移位拥挤距离计算, 较好的平衡解的收敛性和多样性。同时设计了一种动态约束选择策略用于简单任务的实现, 该简单任务只考虑部分约束条件对种群的影响。两个种群通过子代种群进行信息交互。在约束多目标测试函数集 DOC 和 CF、Solomon 基准 VRPTW 数据集以及真实 VRPTW 实例进行对比试验, 实验结果表明基于协同进化框架的约束多目标进化算法具有较好的综合性能。

## 1.4 论文结构安排

本文一共六个章节, 具体安排如下:

第一章 绪论。首先简单介绍了车辆路径问题的研究背景以及研究意义, 然后介绍了当前车辆路径问题模型及相关求解算法的研究现状, 针对一些国内外学者提出的模型及求解算法进行总结, 最后分析本文的研究内容和整体结构安排。

第二章 相关理论基础。首先介绍本文研究中用到的相关约束。然后介绍了多目标优化问题的相关定义。最后对差分进化算法和 NSGAII进行简要的概述。

第三章 带时间窗约束的时变绿色车辆路径问题模型构建。首先对 TDGVRPTW 进行简要的描述, 并给出了相应的时间依赖模型以及碳排放量计算, 然后给出了模型构建的符号表示以及需要考虑的假设条件, 最后着重介绍了 TDGVRPTW 的数学模型。

第四章 基于改进差分进化算法的车辆路径优化。本章首先对改进初始化以及混合变异策略进行了介绍, 然后对算法的整体框架进行了描述, 随后对算法的复杂度以及性能进行了分析, 最后在 Solomon 基准 VRPTW 数据集以及真实 VRPTW 实例上进行仿真实验, 验证了本章所提出的算法在求解提出模型的有效性。

第五章 基于合作协同进化算法的车辆路径优化。本章首先对算法的整体思想进行阐述, 然后给出了基本框架以及具体算法描述, 随后对算法复杂度及性能进行分析, 最后分析本章提出的算法在模型求解中的优势。

第六章 结论与展望。总结本文主要的研究内容, 分析所做工作存在的缺点与不足, 对车辆路径规划未来的研究方向与趋势进行展望。

## 第二章 相关理论基础

### 2.1 车辆路径优化相关理论

#### 2.1.1 时间窗类型

对时间窗的限制表明了客户接受配送服务的时间上下限，如果配送时间早于或晚于这些限制，将会产生一定的惩罚成本，这更符合实际的配送业务场景。

实际情况中，时间窗约束主要包括三种<sup>[8]</sup>：硬时间窗、软时间窗和混合型时间窗。在本研究中，车辆的送达时间均以车辆到达客户点位置为准。在实际物流配送过程中，在一辆车分配的所有客户点中，后到达的客户点受前到达客户点的时间影响，若车辆到达某客户点的时间恰好在要求范围的下限，此时发生车辆卸货等其他情况会导致车辆离开时间有延迟，有可能会造成后续客户送达时间均超出时间窗范围，则会积累较多的时间惩罚成本，即软时间窗约束。因此，软时间窗更适用于实际配送场景，且可扩展的范围更广泛，因此本文研究模型以软时间作为研究。

惩罚成本函数通常由物流企业与客户共同确定，旨在保障双方的利益和企业服务质量。延迟时间和惩罚金额之间的线性关系通常采用分段线性函数来描述，如图 2-1 所示。最大惩罚成本为  $m$ ，客户要求的货品送达时间段为  $[b,c]$ ，该区间的惩罚成本为 0；而客户可接受的送达时间段为  $[a,d]$ ，该区间的时间惩罚成本与时间呈线性关系。

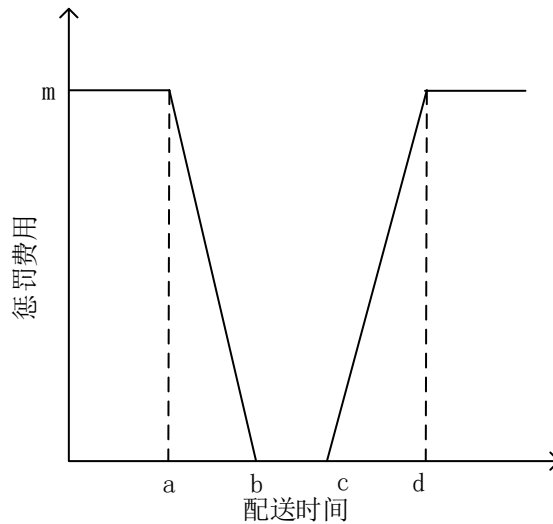


图 2-1 惩罚费用线性函数

根据图 2-1 可知，若配送车辆在时间点  $a$  之前或  $d$  之后送达，则将产生最大的惩罚费用  $m$ ；若在时间段  $[a,b]$  和  $[c,d]$  内送达，则会产生一定的时间惩罚费用；而若在时间段  $[b,c]$  内送达，则不会有任何惩罚费用产生。

#### 2.1.2 时间依赖函数

现有研究对时变环境进行分析时，主要使用时间依赖函数对其进行描述，常见的的时间依赖函数包括如下三种：

### (1)基于行程时间的时间依赖函数

Malandraki 等人<sup>[58]</sup>将车辆行驶时间抽象为阶跃函数, 整个物流配送服务时间均分为多个时间段。一旦车辆在某个时间段出发, 那么就可以唯一确定车辆行驶时间。这个方法过于简单, 违背了先入先出(FIFO)原则。其时间依赖函数如图 2-2 所示。

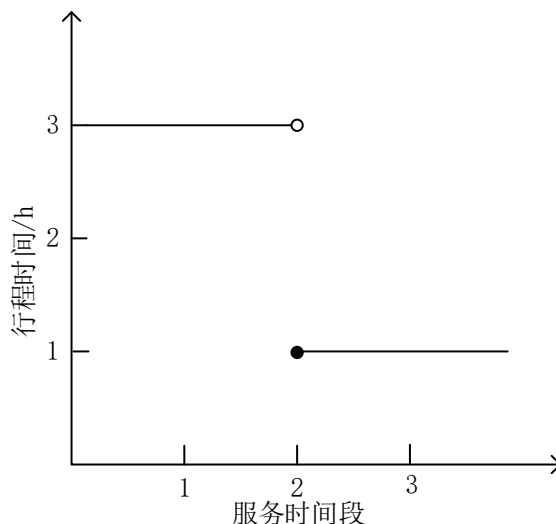


图 2-2 基于行程时间的时间依赖函数

### (2)基于行程速度的时间依赖函数

Ichoua 等人<sup>[59]</sup>使用分段函数描述速度, 使其满足 FIFO 原则, 客户之间的行程时间通过速度进行计算。该时间依赖函数如图 2-3 所示, 将服务时间划分为不同的时间段。

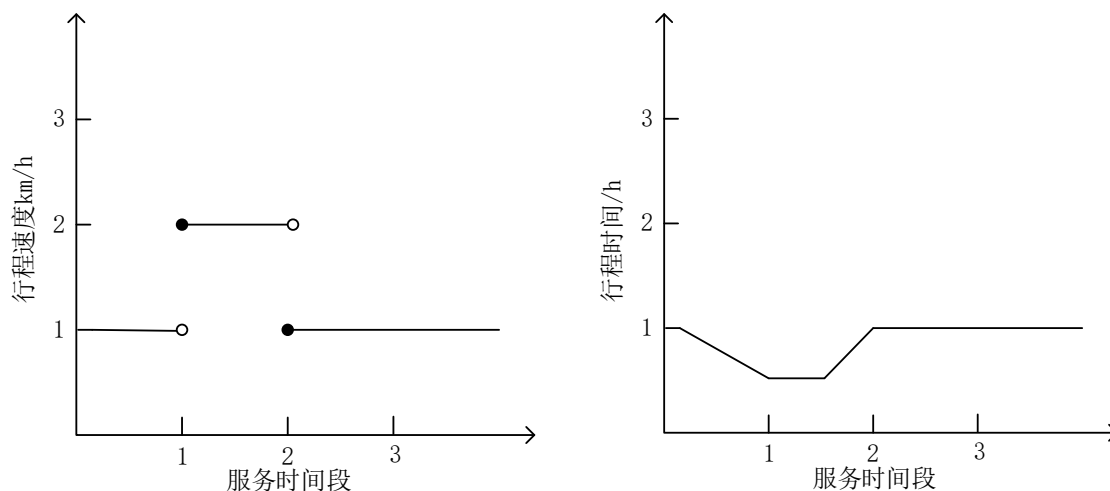


图 2-3 基于行程速度的时间依赖函数

## 2.1.3 碳排放模型

经济快速发展的同时, 由交通运输、基础工业等带来的温室气体排放量不断增加, 其中车辆运输在其中占有重要地位。因此为了缓解温室效应, 众多学者将碳排放量考虑到车辆路径问题中。碳排放与油耗呈正比关系, 因此学者根据这个关系对碳排放进行分析建模, 致力于构建符合实际的油耗计算模型。集中具有代表性的模型如下所示:

## (1)瞬时油耗计算模型

该模型由 Bowyer 等人<sup>[60]</sup>针对短途运输方式提出, 该模型根据车辆载重、牵引力以及油耗率等模拟行驶过程中的单位时间油耗。其计算公式(2-1)所示。

$$f(t) = \begin{cases} \mu_1 + \mu_2 P(t)v + (\mu_3 ma^2 v / 1000), & P(t) > 0 \\ a, & P(t) \leq 0 \end{cases} \quad (2-1)$$

$$P(t) = \sigma_1 + \sigma_2 v^2 + ma / 1000 + gm\omega / 10000 \quad (2-2)$$

其中,  $\mu_1$  为车辆等待时的油耗常数,  $\mu_2$  和  $\mu_3$  分别为单位能量油耗和边际油耗,  $\sigma_1$  和  $\sigma_2$  为阻力和动力。  $a$  为加速度,  $m$  为重量,  $\omega$  为百分比等级,  $v$  为速度。

根据上述模型可以计算, 若总行驶时间为  $T$ , 那么总油耗计算如公式(2-3)所示。

$$F(t) = \int_0^T f(t)dt \quad (2-3)$$

## (2)综合油耗计算模型

该模型由 Barth 等人<sup>[61]</sup>针对重型车辆提出, 比瞬时油耗计算模型多考虑了发动机功率和转速的影响。相对来说, 计算更加符合现实情况。该模型计算如公式(2-4)所示。

$$f = \mu \left[ kN_e V \frac{d}{v} + \gamma \beta dv^2 + \gamma \alpha (m_1 + m_2) d \right] \quad (2-4)$$

其中,  $\mu = \zeta / \kappa \eta$ ,  $\gamma = 1 / 1000 \varepsilon \varpi$ ,  $\alpha = g \sin \theta + g C_r \cos \theta$ ,  $\beta = 0.5 C_d A \rho$ ,  $\zeta$  为燃油空气质量比,  $\varepsilon$  为传动效率,  $\eta$  为转换系数,  $\varpi$  为发动机效率,  $k$  为摩擦因子,  $V$  为排量,  $A$  为迎风面积,  $m_1$  和  $m_2$  为本身车重及载重量,  $C_r$  和  $C_d$  分别为气动和滚动阻尼系数,  $g$  为重量常量,  $\rho$  为空气密度,  $\theta$  为坡度。

## (3)“Bektas-Laporte”油耗计算模型

该模型由 Bektas 等人<sup>[46]</sup>提出, 其具体计算公式如公式(2-5)所示:

$$FC_{ij} = \left[ h_{ij} (\mu + w_{ij}) + \sigma v_{ij}^2 \right] d_{ij} \quad (2-5)$$

其中,  $FC_{ij}$  为由客户  $i$  到客户  $j$  的油耗量,  $h_{ij}$  为与坡度、阻力相关的常量,  $\mu$  为车重量,  $w_{ij}$  为载重量,  $\sigma$  为车型相关常量,  $v_{ij}$  为行驶速度,  $d_{ij}$  为距离长度。

## 2.1.4 相关数据集

本文研究问题的以带时间窗的车辆路径问题为主, 因此本文主要采用 VRPTW 的相关数据集。主要包括 Solomon 基准数据集<sup>[28]</sup>以及真实 MOVRPTW 实例<sup>[62]</sup>。详细介绍如下:

(1)国际通用的标准数据集为 Solomon 基准数据集, 该模型包含三种不同分布类型的数据集, 分别为 R、C 以及 RC, R 代表客户点随机分布, C 代表客户点聚类到确定的区域, RC 则表示混合了聚类和随机分布。每种类型下客户点坐标相同, 但具有不同的时间窗口宽度。数据集详细描述了客户的坐标、时间窗口、服务时间、需求量、配送中心的最大车辆数以及车辆最大载重量, 并假设车辆在配送过程中车速恒定不变, 客户之间、客户与配送中心之间的距离为坐标点的欧氏距离。客户信息示例如表 2-1 所示。

表 2-1 客户信息实例

客户编号	横坐标	纵坐标	需求量	最早服务时间	最晚服务时间	服务时间
0	40	50	0	0	240	0
1	25	85	20	145	175	10
2	22	75	30	50	80	10
3	22	85	10	109	139	10
4	20	80	40	141	171	10
5	20	85	20	41	71	10
6	18	75	20	95	125	10
7	15	75	20	79	109	10
8	15	80	10	91	121	10
9	10	35	20	91	121	10
10	10	40	30	119	149	10

(2)基于真实运输公司数据建立了一个多目标 VRPTW 实例。该数据集通过记录的真实数据建立了客户点之间的行程距离矩阵。该实例含有通过五种不同的时间窗口配置策略（即  $tw0$ 、 $tw1$ 、 $tw2$ 、 $tw3$  和  $tw4$ ）以及三种车辆载重量（即  $d0$ 、 $d1$  和  $d2$ ）的组合而生成的。车辆载重量根据公式  $C = d_{\max} + \delta/100(D - d_{\max})$  计算，三种车辆载重量分别由  $\delta$  取值为  $\{60, 20, 5\}$  时计算得到。其中， $d_{\max}$  代表客户的需求量最大值， $D$  为所有客户的总需求量。此数据集和配置文件可从文献[63]下载。

这五个时间窗口配置类型描述如下：

类型 1. 所有客户全天可用，即整个 8 小时（480 分钟）。

类型 2. 考虑三种客户，即早期客户、中午客户和晚期客户。480 分钟的总时间分为三个相等的部分，即每种客户的 160 分钟时隙。早期客户将在  $[0, 160)$  分钟的时段内得到服务，中午客户将在  $[160, 320)$  分钟的时间段内得到服务；晚期客户将在  $[320, 480]$  分钟的时间间隔内得到服务。

表 2-2 客户信息实例

客户编号	横坐标	纵坐标	需求量	最早服务时间	最晚服务时间	服务时间
0	116.571614	39.792844	0	480	1440	0
1	116.242043	40.07263	207.6	540	720	10
2	116.403595	39.872945	58.63	810	840	10
3	116.186289	40.016361	36.45	780	900	10
4	116.508011	39.826296	25.95	540	600	10
5	116.130997	39.825921	19.8	660	810	10
6	116.292094	39.942186	26.53	690	780	10
7	116.439797	39.833047	60.97	540	660	10
8	116.508963	39.82386	10.75	780	900	10
9	116.378689	39.795018	221.895	540	720	10
10	116.358365	39.922532	25	780	840	10

类型 3. 每个时间窗口的长度减少 30 分钟。因此，早期客户的时间为  $[0, 130]$ ，中期客户的时间为  $[175, 305]$ ，晚期客户的时间段为  $[350, 480]$ 。

类型 4. 每个时间窗口的长度进一步缩短 30 分钟。因此，早期客户的时间间隔为  $[0,$

100], 中期客户的时间为[190, 290], 晚期客户的时间段为[380, 480]。

类型 5. 客户可以属于任何上述时间窗口配置文件

$tw0$  只包含第一种类型的时间窗口,  $tw1$ 、 $tw2$  和  $tw3$  分别包含类型 2、3 和 4,  $tw4$  包含所有类型时间窗口, 每个配置文件中每个时间窗口类型出现的概率相同。

(3) A 公司真实数据中有一家中心仓库作为配送中心, 本文选出其中 100 个客户点对其进行物流配送服务。表 2-2 列出部分客户点的经纬度、服务时间、需求量等信息。

## 2.2 多目标优化相关理论

### 2.2.1 多目标优化问题定义

约束多目标优化问题在许多工程应用中经常出现, 例如路径规划<sup>[63]</sup>, Web 服务器定位<sup>[64]</sup>和作业调度<sup>[65]</sup>等问题。CMOPs(Constrained Multi-objective Optimization Problems)涉及多个互相冲突的目标函数以及不同的约束, 约束可能会改变问题搜索区域, 形成不可行区域。因此求解 CMOP 比无约束多目标问题更加困难。通常, CMOP 定义为<sup>[66]</sup>:

$$\min F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T, \mathbf{x} \in \Omega \quad (2-6)$$

$$s.t. \begin{cases} g_j(\mathbf{x}) \leq 0, j = 1, \dots, q \\ h_j(\mathbf{x}) = 0, j = q+1, \dots, p \end{cases} \quad (2-7)$$

其中  $\mathbf{x}$  是决策空间中的决策变量向量,  $F(\mathbf{x})$  是具有  $m$  个实数函数值的目标向量。地方算法  $g_j(\mathbf{x})$  和  $h_j(\mathbf{x})$  分别是第  $j$  个不等式和等式约束, 其数量分别是  $q$  个和  $p-q$  个。 $\mathbf{x}$  对第  $j$  个约束的约束违反通常计算为:

$$CV_j(\mathbf{x}) = \begin{cases} \max\{0, g_j(\mathbf{x})\}, j = 1, \dots, q \\ \min\{0, |h_j(\mathbf{x})|\}, j = q+1, \dots, p \end{cases} \quad (2-8)$$

总体约束违反值  $\phi(\mathbf{x})$  计算如下:

$$\phi(\mathbf{x}) = \sum_{j=1}^p CV_j(\mathbf{x}) \quad (2-9)$$

对于一个决策变量向量  $\mathbf{x}$ , 只有当整体约束违反值为零时为可行解。对于两个可行解  $\mathbf{x}_1$  和  $\mathbf{x}_2$ , 同时满足  $f_h(\mathbf{x}_1) \leq f_h(\mathbf{x}_2), \forall h \in \{1, \dots, m\}$  以及  $f_g(\mathbf{x}_1) < f_g(\mathbf{x}_2), \exists g \in \{1, \dots, m\}$ , 则称  $\mathbf{x}_1$  支配  $\mathbf{x}_2$ 。当没有其他可行解占主导地位时, 该解被称为 Pareto 最优解。决策空间中的所有 Pareto 最优解构成 Pareto 最优解集(PS)。目标空间中所有 Pareto 最优解对应的目标函数值构成 Pareto Front (PF)。求解 CMOPs 的目标是得到一组收敛良好且分布良好的可行解, 以更好的逼近 PF<sup>[67]</sup>。

### 2.2.2 差分进化算法概述

差分进化(Differential Evolution, DE)算法<sup>[68][69]</sup>作为一种新型的智能算法, 其原理简单、受控参数少、鲁棒性好、易于实现。DE 算法是在遗传算法等进化思想的基础上提出的, 其基本思想即来源于遗传算法, 模拟遗传算法中的杂交、变异、复制来设计算子。



与其他基于种群的随机算法类似, DE 算法的过程主要包括四个步骤: 种群初始化、变异、交叉和选择。如图 2-4 所示。

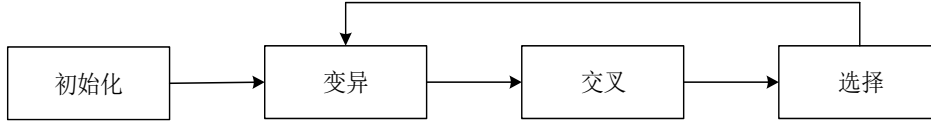


图 2-4 DE 算法的过程

### (1) 种群初始化

在解空间内随机均匀地生成  $NP$  个个体, 每个个体均为一个  $D$  维向量:

$$\mathbf{x}_{i,j}^0 = \mathbf{l}_j + \text{rand}(0,1) * (\mathbf{u}_j - \mathbf{l}_j), \quad i = 1, 2, \dots, NP \quad (2-10)$$

其中,  $\mathbf{x}_{i,j}^0$  表示第 0 代的第  $i$  个个体的第  $j$  个变量,  $\mathbf{l}$  和  $\mathbf{u}$  代表问题所给出的决策空间的最小向量和最大向量。即, 每个个体代表解空间内的某一个解, 而基因代表解的各个分量。

### (2) 变异算子

为了保证多样性, 变异中需要选择的随机粒子不应与当前更新粒子相同。变异策略通常有以下 5 种:

$$\text{DE/rand/1: } \mathbf{v}_i = \mathbf{x}_{r1} + F(\mathbf{x}_{r2} - \mathbf{x}_{r3})$$

$$\text{DE/rand/2: } \mathbf{v}_i = \mathbf{x}_{r1} + F[(\mathbf{x}_{r2} - \mathbf{x}_{r3}) + (\mathbf{x}_{r4} - \mathbf{x}_{r5})]$$

$$\text{DE/best/1: } \mathbf{v}_i = \mathbf{x}_{best} + F(\mathbf{x}_{r2} - \mathbf{x}_{r3})$$

$$\text{DE/best/2: } \mathbf{v}_i = \mathbf{x}_{best} + F[(\mathbf{x}_{r2} - \mathbf{x}_{r3}) + (\mathbf{x}_{r4} - \mathbf{x}_{r5})]$$

$$\text{DE/current-to-best/1: } \mathbf{v}_i = \mathbf{x}_i + F[(\mathbf{x}_{best} - \mathbf{x}_i) + (\mathbf{x}_{r1} - \mathbf{x}_{r2})]$$

其中,  $\mathbf{v}_i$  为变异向量,  $r1$ 、 $r2$ 、 $r3$ 、 $r4$ 、 $r5$  是为种群中的五个互不相等的随机数,  $\mathbf{x}_{best}$  为当前群体中最好的个体。  $F$  为缩放因子, 用于控制差分向量的影响, 通常设置为 0.5。

### (3) 交叉算子

引入交叉算子可以增强种群的多样性。在第  $t$  次迭代中, 对于每个个体和它所生成的变异向量进行交叉, 具体的说就是对每一个等位基因 (分量) 按照一定的概率选择变异向量来生成交叉向量。

$$\mathbf{c}_{i,j}^t = \begin{cases} \mathbf{v}_{i,j}^t, & \text{rand}_{i,j} < CR \text{ or } j = j_{rand} \\ \mathbf{x}_{i,j}^t, & \text{otherwise} \end{cases} \quad (2-11)$$

其中,  $\mathbf{c}$  为交叉向量,  $CR$  是交叉概率。  $CR$  的值越大, 发生交叉的概率越大。

### (4) 选择算子

该过程根据适应度函数的值从第  $t$  次迭代中每个个体的交叉向量和原始向量中选择出适应度更高的作为下一代。

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{c}_i^t, & f(\mathbf{c}_i^t) < f(\mathbf{x}_i^t) \\ \mathbf{x}_i^t, & \text{otherwise} \end{cases} \quad (2-12)$$

差分进化算法主要流程如算法 2.1 所示。

---

 算法 2.1 DE 算法的主要流程
 

---

 输入：种群大小  $N$ 、最大迭代次数  $MaxGen$ 

 输出：种群  $P_t$ 

```

1   $t \leftarrow 1$ ;
2  种群初始化  $P_t = \{\mathbf{x}_1^t, \dots, \mathbf{x}_N^t\}$ ;
3  while  $t \leq MaxGen$  do
4    for  $i = 1:N$  do;
5      根据变异以及交叉算子生成子代  $\mathbf{c}_i^t$ ;
6      对  $\mathbf{c}_i^t$  和  $\mathbf{x}_i^t$  进行贪心选择进入下一代种群  $P_{t+1}$ ;
7    end
8     $t \leftarrow t + 1$ 
9  end
10 return  $P_t$ ;

```

---

## 2.2.3 NSGAI算法概述

NSGAI是由 Deb 等人<sup>[70]</sup>设计的经典进化多目标优化算法。算法 2.2 给出了 NSGAI 的主要流程。

## (1)快速非支配排序

计算每个个体  $\mathbf{x}_i$  的被支配个数  $n_i$  和被该个体支配的个体集合  $S_i$ ，将种群中  $n_i = 0$  的个体放入第一层级集合  $F_1$  中，并对被该等级个体支配的个体执行  $n_i = n_i - 1$ ，如果  $n_i = 0$  则放入下一个层级集合  $F_2$  中，一直执行类似操作直到种群所有个体全部完成。

---

 算法 2.2 NSGAI算法的主要流程
 

---

 输入：种群大小  $N$ 、最大迭代次数  $MaxGen$ 

 输出：种群  $P_t$ 

```

1   $t \leftarrow 1$ ;
2  随机初始化  $P_t = \{\mathbf{x}_1^t, \dots, \mathbf{x}_N^t\}$ ;
3  while  $t \leq MaxGen$  do
4    根据二元锦标赛从种群中选择父代个体 Parent;
5    根据交叉以及变异算子生成子代 Off;
6     $P_t \leftarrow P_t \cup \text{Off}$ ;
7    对  $P_t$  进行快速非支配排序;
8    根据精英选择策略选择  $N$  个个体进入下一代  $P_{t+1}$ ;
9     $t \leftarrow t + 1$ ;
10 end
11 return  $P_t$ ;

```

---

## (2)拥挤度排序

NSGAI<sub>II</sub>为了保证种群个体分布均匀,引入了拥挤距离进行排序,保证了种群的多样性。在每一层非支配个体中,其拥挤距离计算公式如下:

$$d_i = \sum_{j=1}^m \frac{f_j[i+1] - f_j[i-1]}{f_j^{\max} - f_j^{\min}} \quad (2-13)$$

其中,  $d_i$  为个体  $x_i$  的拥挤距离,  $m$  为目标个数,  $f_j[i+1]$  和  $f_j[i-1]$  分别为个体  $x_i$  在第  $j$  个目标上的前后个体目标值。

### (3)精英选择策略

NSGAI<sub>II</sub>的选择策略可以描述为首先混合父代和子代种群,然后根据快速非支配排序进行分层,然后从低到高依次将该层个体放到下一代种群中,直到某一层个体的加入超出种群大小  $N$ ,此时将该层个体根据拥挤距离进行排序,依次放入下一代种群中,直至种群大小为  $N$ 。

## 2.2.4 多目标优化性能评价指标

多目标优化算法的优劣往往需要相关的指标进行衡量,本节对常用的指标进行介绍,包括反向迭代距离(Inverted Generational Distance,IGD)<sup>[71]</sup>、超体积(Hypervolume,HV)<sup>[72]</sup>。

### (1)IGD

IGD 指标用来表示获得的 PF 与真实 PF 之间欧式距离和的平均值。IGD 越小,代表算法性能越好。IGD 计算公式如下所示:

$$IGD(O, P^*) = \frac{\sum_{q \in P^*} d(q, O)}{|P^*|} \quad (2-14)$$

其中,  $O$  表示获得 PF,  $P^*$  表示真实 PF,  $d(q, O)$  表示从  $q$  与  $O$  之间解的欧氏距离。

### (2)HV

HV 指标用来评估获得的 PF 与其参考点的超体积大小,主要用来验证解在目标空间的分布性。值越大性能越好。HV 计算公式如下所示:

$$HV(P, y^*) = \text{volume} \cup_{x \in P} v(x, y^*) \quad (2-15)$$

其中  $P$  是求解得到的 PF,  $y^*$  是测试函数的参考点,  $v(x, y^*)$  是指一个解与  $y^*$  组成的超立体体积。

## 2.3 本章小结

本章主要介绍车辆路径规划问题和多目标优化的相关基础理论。首先介绍了车辆路径规划问题中的时间窗类型、时间依赖函数、碳排放计算模型以及相关数据集。然后介绍了多目标优化的相关理论,给出了多目标优化问题的定义,并对差分进化算法和 NSGAI<sub>II</sub>进行了简要的概述,最后给出了本文设计的评价指标及相应的计算公式以便对实验结果进行分析。

## 第三章 带时间窗约束的时变绿色多目标车辆路径问题模型构建

### 3.1 带时间窗约束的时变绿色多目标车辆路径问题

实际生活中许多问题都可以归结为VRPTW来处理，如钢铁厂编制热轧带钢轧制计划问题实际上就是一个VRPTW问题。物流运输业中也普遍存在这样的问题：如邮政投递，飞机、火车及公共汽车的调度等。物流运输业在经济发展中必不可缺，与此同时，随着人们对绿色低碳经济的关注越来越多，2009年我国承诺在2020年将单位GDP碳排放量降低40%到50%，欧盟在2012年初对碳排放量征收税务。我国在2020年超额完成2020年碳减排目标，于2021承诺在2025年初步形成绿色低碳循环发展的经济体系，将单位国内生产总值二氧化碳排放比2020年下降19%。因此为了我国战略目标的达成，减少车辆运输过程中造成的污染，在车辆路径问题中对碳排放量的关注仍需持续加强。

车辆行驶过程中二氧化碳的排放往往取决于多个因素，如车辆本身的各种参数、车辆的速度以及载重量等。并且在现实生活中，往往需要同时考虑两个及以上优化目标以满足配送公司以及客户的需求，这些目标之间的关系复杂，往往一个目标的优化会导致另一个目标的优化程度降低。可以发现大部分模型只考虑了一个或两个目标，考虑多个目标的较少，对碳排放影响因素的考虑也较少。因此本章提出在时变环境下，考虑碳排放的总运输成本、总运输时间和车辆数的带时间窗约束的绿色多目标车辆路径问题模型(TDGVRPTW)，其中碳排放量的计算综合考虑了速度、载重以及距离等影响因素。

### 3.2 问题描述

考虑时间窗约束的车辆路径规划问题可以定义如下：设 $G = \{V, A\}$ ，其中 $V = \{0, \dots, N\}$ 是顶点集合，顶点0为配送中心，其他顶点为客户， $A$ 是弧的集合。配送中心使用具有容量 $Cap$ 的同种运输车型的车队来服务这些客户。每个客户都有一个 $m_i$ 的需求以及一个时间窗口 $[b_i, e_i]$ 。服务时间窗口约束表示车辆应该不晚于最晚配送时间 $e_i$ 后的最大允许延迟时间 $md$ 后到达客户 $i$ 的位置完成配送任务，且在早于最早配送时间之前到达客户点时必须需要等待直到 $b_i$ 开始配送任务。每个客户 $i$ 都有一个配送时间 $s_i$ ， $s_i$ 是每辆车到达客户位置开始配送任务后交付一次货物所花费的时间。配送中心的时间窗为 $[0, e_0]$ 。顶点 $i$ 与顶点 $j$ 之间的距离为 $d_{ij}$ ，行驶时间为 $t_{ij}$ 。该问题的目标是求解一组路线 $R = \{r_1, \dots, r_M\}$ 在满足约束的情况下尽可能优化所求解目标函数的成本。

结合现实情况，在实际的物流配送中，车辆在道路上行驶的速度往往与出发时间以及道路状况相关，如车辆在拥堵时段只能维持一个很低的速度。在非拥堵时段则可以以一个较高的速度行驶。本文研究考虑时间窗约束的时变绿色物流配送问题，可以描述为：对于单个配送中心，存在若干从配送中心出发的车辆，将若干物流订单送往对应的客户，这些客户限制了具体配送的时间窗范围，若车辆早于或者晚于时间窗到达则需要对其进行惩罚，其中车辆的速度因为不同时段的道路拥堵情况而不同，在满足以上约束及客户需求下，合理规划物流配送线路，尽可能优化所求解目标函数的成本。

与传统的带时间窗约束的车辆路径规划问题相比，TDGVRPTW 增加了对车辆的载重变化和速度变化的考虑，更加符合现实情况。碳排放量的计算与车辆的载重、速度以及行驶距离有关，因此行驶过程中的载重变化和速度变化将对碳排放量的计算造成影响。而且由于速度的不同所以车辆行驶相同距离所需要花费的时间也随之不同，这将对行驶时间造成一定的影响。在 TDGVRPTW 中，时变条件采用满足 FIFO 的基于行程速度的时间依赖函数，碳排放量采用同时考虑距离、速度以及载重的 Bektas-Laporte 油耗计算模型进行近似计算。

### 3.3 模型构建

#### 3.3.1 模型假设

由于现实情况复杂多变，因此为了更好的开展研究，在带时间窗约束的车辆路径问题模型中做出如下假设：

- (1) 只有一个配送中心，且位置已知。
- (2) 每个客户必须且只能访问一次，需求不可拆分。
- (3) 客户希望接受访问的时间窗口已知，且延迟时间不得大于最大允许延迟时间，早于时间窗口到达则需要等待。
- (4) 客户货物的重量等于需求量
- (5) 车辆有承载量限制、无最大运输行程限制
- (6) 使用车辆数有上限、车辆均采用同一种运输车型
- (7) 车辆必须从配送中心出发、最后必须返回配送中心
- (8) 车辆行驶过程中不考虑坡度的影响
- (9) 车辆行驶过程中无停顿，如等待红绿灯等时间花费

#### 3.3.2 符号定义

模型相关符号及意义表示如下：

$V$ ：顶点集合  $\{0, \dots, n\}$ ，0 代表配送中心， $\{1, \dots, n\}$  为客户编号

$A$ ：弧的集合

$x_{ijk}$ ：决策变量，如果车辆  $k$  驶过弧  $(i, j)$ ，则为 1，否则为 0

$Cap$ ：车辆的载重量

$WV$ ：车辆的重量

$c_d$ ：单位距离行驶成本

$c_t$ ：单位延迟时间惩罚成本

$c_v$ ：车辆固定成本

$NV$ ：允许使用的最大车辆数

$m_i$ ：客户  $i$  的需求量

$b_i$ ：客户  $i$  的最早配送时间

$e_i$ ：客户  $i$  的最晚配送时间

- $md$ : 客户的最大允许早到或延迟时间  
 $s_i$ : 客户  $i$  的配送时间  
 $r_j$ : 车辆  $j$  的行驶路线  
 $N_j$ : 车辆  $j$  服务的客户数量  
 $T_j$ : 车辆  $j$  的行驶时间  
 $CE$ : 碳排放量计算公式  
 $FC$ : 油耗计算公式  
 $h(i, j)$ : 车辆  $j$  的行驶路线中的第  $i$  个节点的客户编号  
 $a_{h(i, j)}$ : 车辆  $j$  到达路线中第  $i$  个节点的时间  
 $l_{h(i, j)}$ : 车辆  $j$  离开路线中第  $i$  个节点的时间  
 $w_{h(i, j)}$ : 车辆  $j$  在路线中第  $i$  个节点的等待时间  
 $t_{h(i-1, j)h(i, j)}$ : 车辆  $j$  在路线中从第  $i-1$  个节点到第  $i$  个节点所需的行驶时间  
 $delay_{h(i, j)}$ : 车辆  $j$  在路线中第  $i$  个节点的延迟时间  
 $weight_{h(i, j)}$ : 车辆  $j$  在路线中前往第  $i$  个节点时的重量  
 $L$ : 时间段数量  
 $v_i$ : 在第  $i$  个时间段的车辆行驶速度  
 $T^i$ : 第  $i$  个时间段的开始时间  
 $\alpha$ : 碳排放量转换系数

### 3.3.3 行驶时间计算

---

算法 3.1  $ride(t_{begin}, d)$

---

输入: 出发时间  $t_{begin}$ 、行驶距离  $d$

输出:  $t$

- 1 确定出发时间所在时间段  $k$ , 即满足  $t_{begin} \in [T^k, T^{k+1}]$ ;
  - 2  $d_{imp} = (T^{k+1} - t_{begin})v_k, t = 0$ ;
  - 3 **while**  $d_{imp} \leq d$  **do**
  - 4      $t = t + d_{imp} / v_k$ ;
  - 5      $d = d - d_{imp}$ ;
  - 6      $k = k + 1$ ;
  - 7      $d_{imp} = (T^{k+1} - T^k)v_k$ ;
  - 8 **end**
  - 9  $t = t + d_{imp} / v_k$ ;
  - 10 **return**  $t$ ;
- 

本文采用的时变模型中, 将配送时间分成了  $L$  段, 每个时间段的速度可能不同, 车辆在行驶过程中有很大概率跨越多个时间段。客户  $i$  和客户  $j$  之间的行程时间计算函数

$ride(t_{begin}, d)$  如算法3.1所示。

### 3.3.4 TDGVRPTW 模型

基于上述假设和定义，为求解带时间窗约束的车辆路径问题，建立如下混合整数规划数学模型：

该模型考虑的三个目标分别为总运输成本、总运输时间和车辆数。

车辆的总运输成本包含碳排放成本、时间延迟成本、车辆固定成本以及行驶成本，计算公式为：

$$f_1 = \sum_{j=1}^M \sum_{i=0}^{N_j} (CE + c_t \text{delay}_{h(i,j)} + c_d d_{h(i,j)h(i+1,j)}) + c_v M \quad (3-1)$$

配送过程中所有车辆的总运输时间包括了车辆的行驶时间和等待时间，计算公式为：

$$f_2 = \sum_{j=1}^M T_j \quad (3-2)$$

完成此次配送任务需要的车辆数为：

$$f_3 = |R| = M \quad (3-3)$$

该模型的优化目标可以概括为：

$$\min f = \{f_1, f_2, f_3\} \quad (3-4)$$

模型包含了各种约束条件，如容量约束、时间窗约束等。约束条件的具体定义如下：

保证每个客户点只被一辆车服务：

$$\sum_{k=1}^M \sum_{j \in V} x_{ijk} = 1, i \in V \setminus \{0\} \quad (3-5)$$

每辆车最多完成一次配送任务：

$$\sum_{j \in V} x_{0,jk} \leq 1, \forall k \in \{1, \dots, M\} \quad (3-6)$$

每辆车驶入一个客户节点后必须满足依然从该客户节点驶出：

$$\sum_{i \in V} x_{ijk} - \sum_{j \in V} x_{jik} = 0, j \in V, \forall k \in \{1, \dots, M\} \quad (3-7)$$

车辆从配送中心驶出，必须返回配送中心：

$$\sum_{j \in V} x_{0,jk} = \sum_{i \in V} x_{i0k}, \forall k \in \{1, \dots, M\} \quad (3-8)$$

车辆装载的货物不能超过载重量：

$$\text{weight}_{h(1,j)} \leq \text{Cap}, \forall j \in \{1, \dots, M\} \quad (3-9)$$

所有车辆都应在配送中心时间窗口结束前返回配送中心：

$$a_{h(N_j+1,j)} \leq e_0, \forall j \in \{1, \dots, M\} \quad (3-10)$$

所有路线中车辆延迟时间不能超过最大允许延迟时间：

$$\text{delay}_{h(i,j)} \leq md, \forall i \in \{1, \dots, N_j\}, \forall j \in \{1, \dots, M\} \quad (3-11)$$

车辆的速度与出发的时间相关联，因此每条路线的行驶时间都需要动态的计算，碳排放量的计算也与车辆的载重有关，因此车辆的重量以及路线的行驶时间计算如下：

为了方便计算增加了每条路线的起始点和结束点都为配送中心：

$$h(0, j) = h(N_j + 1, j) = 0 \quad (3-12)$$

等待时间  $w_{h(i,j)}$  的计算公式如下:

$$w_{h(i,j)} = \begin{cases} 0, & \text{if } a_{h(i,j)} \geq b_{h(i,j)} - md \\ b_{h(i,j)} - a_{h(i,j)}, & \text{otherwise} \end{cases} \quad (3-13)$$

离开时间  $l_{h(i,j)}$  的计算公式如下:

$$l_{h(i,j)} = a_{h(i,j)} + w_{h(i,j)} + s_{h(i,j)} \quad (3-14)$$

到达时间  $a_{h(i,j)}$  的计算公式如下:

$$a_{h(i,j)} = l_{h(i-1,j)} + t_{h(i-1,j)h(i,j)} \quad (3-15)$$

$t_{h(i-1,j)h(i,j)}$  的计算跟行驶距离、行驶速度以及出发时间有关, 计算方式如下:

$$t_{h(i-1,j)h(i,j)} = \text{ride}(l_{h(i-1,j)}, d_{h(i-1,j)h(i,j)}) \quad (3-16)$$

延迟时间  $\text{delay}_{h(i,j)}$  的计算公式如下:

$$\text{delay}_{h(i,j)} = \begin{cases} 0, & \text{if } a_{h(i,j)} \leq e_{h(i,j)} \\ a_{h(i,j)} - e_{h(i,j)}, & \text{otherwise} \end{cases} \quad (3-17)$$

每条路线上车辆的总行驶时间:

$$T_j = \sum_{i=0}^{N_j} (t_{h(i,j)h(i+1,j)} + w_{h(i,j)} + s_{h(i,j)}) \quad (3-18)$$

每辆车的总重量变化公式如下:

$$\text{weight}_{h(i,j)} = \text{weight}_{h(i-1,j)} - m_{h(i-1,j)} \quad (3-19)$$

碳排放量计算公式如下:

$$CE(v, g, d, l) = \alpha * FC(v, g, d, l) \quad (3-20)$$

### 3.4 本章小结

本章首先对带时间窗约束的考虑载重量、速度和距离的绿色车辆路径问题进行描述, 说明了提出的模型与经典的带时间窗约束的车辆路径问题的区别。然后对模型做出一些必要的假设以及使用符号的定义, 将考虑碳排放的总成本、最小行驶时间以及最少车辆数作为模型的目标, 最终给出解决带时间窗约束的考虑载重量、速度和距离的绿色车辆路径问题的数学模型。





## 第四章 基于改进差分进化算法的车辆路径优化

VRPTW 作为 VRP 的变种能够模拟各种领域的实际应用，同时也是一个多目标优化问题，其目标函数根据不同应用场景而不同。例如，对于货物运输也考虑更多的是行驶距离作为关键目标，因为这关系着油耗成本。快递运输则需考虑运输时间的长短以提高客户满意度。第三章构建了一种在时变环境下以考虑碳排放量的最小运输成本、最小行驶时间以及最小车辆数为目标的 TDGVRPTW 模型。对于多目标 TDGVRPTW 模型的求解需要在满足约束的同时兼顾多目标优化，目标数量的增加给算法求得优质解带来了困难。为了求得更好的解集，本章提出一种改进初始化、变异和选择策略的多目标差分进化算法(Improved Initialization Mutation and Selection strategy based Differential evolution, IMSDE)用于求解该模型。在该改进算法中，初始化策略将 Tent 混沌映射和反向学习相结合，增强了初始种群的多样性和分布性。变异策略中结合两种变异算子对多样性和收敛性进行更好的平衡，并设计了一种修补算子避免种群聚集在边界上，降低种群的多样性。自适应选择策略使得一些潜在优质解能够保留，避免种群陷入局部最优。

### 4.1 基于改进初始化和变异选择策略的差分进化算法

#### 4.1.1 反向混沌初始化

对于进化算法来说，初始种群均匀分布在决策空间中可以提高算法的求解精度和收敛速度，增大算法的搜索范围，对获得最终解的质量起到了关键性的作用。提出一种基于 Tent 混沌映射<sup>[73]</sup>和反向学习<sup>[74]</sup>的种群初始化方法。

Tent 混沌映射函数公式如下：

$$Z_{k+1} = \begin{cases} Z_k / \beta & 0 < Z_k \leq \beta \\ (1 - Z_k) / (1 - \beta) & \beta < Z_k \leq 1 \end{cases} \quad (4-1)$$

Tent 混沌映射具有良好的分布性和随机性。为了进一步增加种群的多样性，扩大搜索区域的范围，增强算法的全局勘探能力，提高发现最优解的概率，引入反向学习策略处理映射后的种群得到初始种群。反向学习计算公式如下：

$$Y_i' = l + u - rand(0,1) * Y_i \quad (4-2)$$

其中，根据文献[73]  $\beta$  取 0.5， $l$  和  $u$  代表问题所给出的决策空间的最小向量和最大向量， $Y_i$  表示种群中第  $i$  个个体向量。采用 Tent 混沌映射和反向学习能够加快算法收敛速度和效率，能够很快地发现最优解。

生成初始种群具体步骤如下：首先，生成  $N$  个随机数并将其带入公式(4-1)进行 Tent 混沌映射，映射后生成初始种群；然后，将初始种群带入公式(4-2)中生成反向种群，最终得到大小为  $2N$  的初始种群，并对初始种群进行选择。

图 4-1 给出了经过 Tent 混沌映射和反向学习后得到的初始种群分布以及随机初始化生成的初始种群分布图，可以看出改进的初始化方法使种群在决策空间中分布的更加均匀，将提高发现最优解的概率。示例中假设种群大小  $N$  为 100，个体变量维度  $D$  为 2。

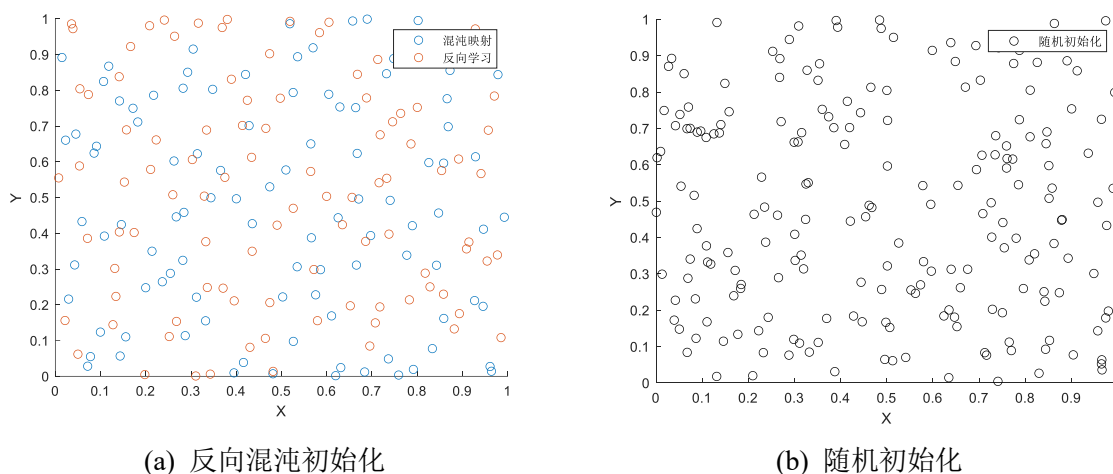


图 4-1 初始种群分布图

#### 4.1.2 变异策略及修补算子

一般来说, 求解 CMOPs 需要考虑多样性和收敛性之间的平衡以及目标函数与约束之间的平衡。为了实现这两个平衡, 综合两种流行 DE 算子 DE/rand-to-best/1/bin 和 DE/current-to-rand/1 实现平衡, 这两个算子已经成功应用于求解约束多目标问题<sup>[75][76]</sup>。其中 DE/rand-to-best/1/bin 如公式(4-3)和(4-4)所示, 该算子采用了最佳个体的信息, 能够加强种群的收敛性。DE/current-to-rand/1 如公式(4-5)所示, 当前个体使用了随机选择个体的信息, 因此可以保持多样性。

DE/rand-to-best/1/bin:

$$\mathbf{y}_i = \mathbf{x}_{r_1} + rand(0,1)(\mathbf{x}_{best} - \mathbf{x}_{r_1}) + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (4-3)$$

$$\mathbf{u}_{i,d} = \begin{cases} y_{i,d}, & \text{if } (rand(0,1) \leq CR \text{ or } d = d_{rand}) \\ x_{i,d}, & \text{otherwise} \end{cases}, d = 1, \dots, D. \quad (4-4)$$

DE/current-to-rand/1:

$$\mathbf{u}_i = \mathbf{x}_{r_1} + rand(\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \quad (4-5)$$

其中,  $i = 1, \dots, N$ ,  $\mathbf{y}_i = (y_{i,1}, \dots, y_{i,D})^T$  和  $\mathbf{u}_i = (u_{i,1}, \dots, u_{i,D})^T$  分别代表第  $i$  个个体的差分向量和实验向量。 $r_1, r_2, r_3$  为互不相等的整数,  $\mathbf{x}_{best}$  为种群中的最优个体。 $F$  是用于缩放差分向量的比例因子,  $CR$  为交叉控制参数。值得注意的是, DE/current-to-rand/1 没有进行二项式交叉, 因此具有旋转不变性<sup>[77]</sup>。

在该策略中, 两个算子被选择的概率相等, 以实现收敛性和多样性的平衡。根据文献 [75][76] 的建议,  $F$  和  $CR$  分别从比例因子池  $\{0.6, 0.8, 1.0\}$  和交叉控制参数池  $\{0.1, 0.2, 1.0\}$  中随机选取。最优个体  $\mathbf{x}_{best}$  的选取关系到目标函数与约束条件之间的平衡, 按照如下规则进行选取。当所有个体都为不可行个体时, 具有最小约束违反的个体作为最优个体, 以增强关于约束条件的收敛性。当种群中含有可行个体时, 通过对可行个体进行非支配排序, 从第一层非支配解上随机选择一个个体作为最优个体。通过上述过程, 目标函数与约束条件之间可以达到一定的平衡。

由于，变异后的向量个体可能是超出解空间范围的，即变异产生的个体不满足边界条件，落在解空间以外。因此对不可行解进行修补<sup>[78]</sup>是进化算法必须有的操作。常见的修补方式如公式(4-6)所示，但是这样的修补方式会导致种群聚集在边界上，降低种群的多样性。

$$Y_i' = \text{Min}(\text{Max}(Y_i', l), u) \quad (4-6)$$

本节设计了一种基于平均向量的修补算子对越界的个体进行处理，以避免种群聚集在解空间边界上，保持种群的多样性。以经典算子 DE/rand/2 为例，具体流程如下。

差分向量生成成为：

$$y_i = x_{r_1} + F \cdot [(x_{r_2} - x_{r_3}) + (x_{r_4} - x_{r_5})] \quad (4-7)$$

如果差分向量  $y_i$  超出解空间范围，则进行如下处理方式：

首先，计算平均向量：

$$x_{avg} = \frac{x_{r_1} + x_{r_2} + x_{r_3} + x_{r_4} + x_{r_5}}{5} \quad (4-8)$$

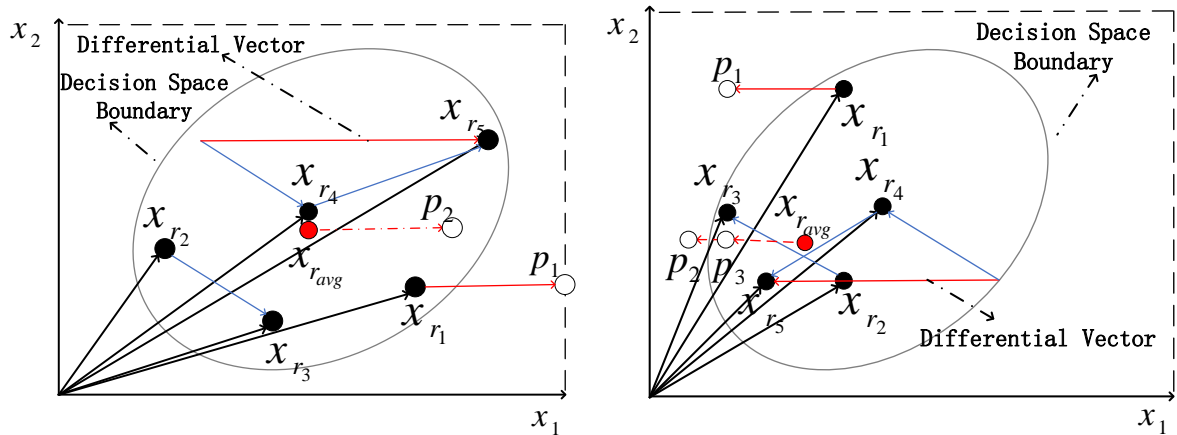
然后，按照如下方式进行第二次变异：

$$y_{i,j} = x_{r_{avg},j} + F \cdot [(x_{r_{2,j}} - x_{r_{3,j}}) + (x_{r_{4,j}} - x_{r_{5,j}})] \quad (4-9)$$

如果经过第二次变异后个体仍然越界，则按照以下方式进行修补：

$$y_{i,j} = \begin{cases} \min\{l_j, y_{i,j} + \text{rand}(0,1) * y_{i,j}\}, & y_{i,j} < l_j \\ \max\{u_j, y_{i,j} - \text{rand}(0,1) * y_{i,j}\}, & y_{i,j} > u_j \end{cases} \quad (4-10)$$

其中， $y_{i,j}$  表示  $i^{\text{th}}$  个个体在  $j^{\text{th}}$  维上的值。



(a) 修补后落在解空间内

(b) 修补后落在解空间外

图 4-2 修补操作示意图

相较于传统修补方式，本节提出的平均修补算子改善了种群的分布性，避免了种群密集分布在解空间边界，增强了种群的多样性。图 4-2 给出了对突变个体进行合法化的说明，其中实心圆点表示差分生成的 5 个个体，空心圆点表示变异生成的个体，红色实心圆点表示 5 个个体的平均个体，图 4-2(a) 举例说明了变异产生的个体  $p_1$  超出解空间范围，通过公式(4-8)计算参与变异的个体的平均向量  $x_{avg}$ ， $x_{avg}$  落在 5 个初始个体之间，

将  $\mathbf{x}_{avg}$  按照公式(4-9)进行二次变异, 得到个体  $\mathbf{p}_2$ ,  $\mathbf{p}_2$  在解空间内, 保证了解空间个体的多样性。图 4-2(b)中首次变异个体  $\mathbf{p}_1$  超出解空间边界, 对其进行上述二次变异得到个体  $\mathbf{p}_2$  仍在解空间外, 此时根据公式(4-10)对  $\mathbf{p}_2$  进行修补, 在解空间内得到  $\mathbf{p}_3$ , 保证了种群的多样性。

通过上述过程, 两种变异算子的混合策略实现了收敛性和多样性的平衡与目标函数与约束条件之间的平衡, 并且在变异结束后对非法解使用修补算子进行修补, 这样也极大地保持了种群的多样性, 改善了种群的分布情况, 避免种群聚集在边界。算法 4.1 给出了该变异策略的伪代码表示。

---

#### 算法 4.1 变异策略

---

输入: 种群大小  $N$ 、种群  $P_t = \{\mathbf{x}'_1, \dots, \mathbf{x}'_N\}$

输出: 子代种群  $O_t$

```

1   $O_t \leftarrow \emptyset$ ;
2  计算每个个体  $\mathbf{x}'$  的约束违反  $CV(\mathbf{x}')$ ;
3  for  $i = 1: N$  do
4      从  $\{0.6, 0.8, 1.0\}$  随机选择一个值作为  $F$ , 从  $\{0.1, 0.2, 1.0\}$  中随机选择一个值作为  $CR$ ;
6      if  $rand < 0.5$  then
7          if  $P_t$  中没有可行解 then
8              选择  $CV(\mathbf{x}')$  最小的个体作为  $\mathbf{x}_{best}$ ;
9          else
10             根据 Pareto 支配关系进行排序得到第一层非支配解;
11             从第一层解中随机选择一个个体作为  $\mathbf{x}_{best}$ ;
12         end
13         从  $P_t$  中选择  $\mathbf{x}'_{r_1}, \mathbf{x}'_{r_2}$  and  $\mathbf{x}'_{r_3}$ , 根据公式(4-3)和(4-4)产生新个体  $\mathbf{u}'_i$ ;
15     else
16         从  $P_t$  中选择  $\mathbf{x}'_{r_1}, \mathbf{x}'_{r_2}$  and  $\mathbf{x}'_{r_3}$ , 根据公式(4-5)产生新个体  $\mathbf{u}'_i$ ;
18     end
19     if  $\mathbf{u}'_i$  超出边界
20         根据公式(4-9)进行修补;
21         if  $\mathbf{u}'_i$  依然超出边界
22             根据公式(4-10)进行二次修补;
23         end
24      $O_t \leftarrow O_t \cup \mathbf{u}'_i$ ;
25 end

```

---

#### 4.1.3 自适应选择策略

在约束多目标优化中, 常用的选择方法<sup>[79]</sup>为: 首先, 将种群按照约束支配原则分为不同的非支配等级。第二, 选中前几个等级中的所有个体, 直到不能容纳更多的等级。

第三，根据拥挤距离对排名最后的个体进行排序，只选择拥挤距离较小的个体。然而，可能存在个体在非支配排序中处于前级和处于后级的可取程度相反。在进化的前期，获得的 PF 是不完整的。最后几个级别中的某些元素可以填补这一空缺。如图 5-3 所示， $Rank_1$  的拥挤区域中的一些个体是不可取的，而  $Rank_2$  的边界个体是潜在的解。这些潜在的解决方案可能会演变为  $Rank_1$  的稀疏区域。

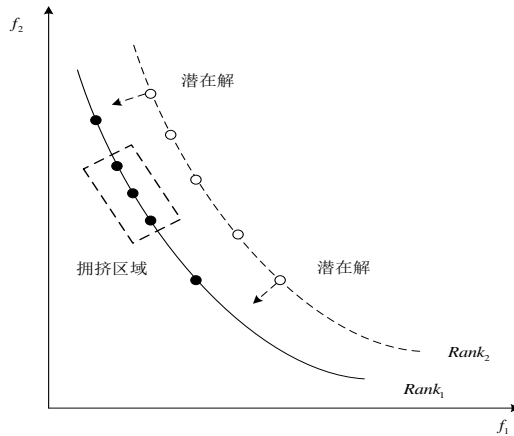


图 4-3 传统选择算子的缺点示意图

为了删除前级中的不良解，并在后级中选择潜在解，在新的选择方法中，只选择了一定百分比的前级元素。为了使获得的 PF 随着迭代次数的增加变得更加完整，所选个体的比率随迭代次数的变化而变化。它们的关系如图 4-4 和公式(4-11)所示：

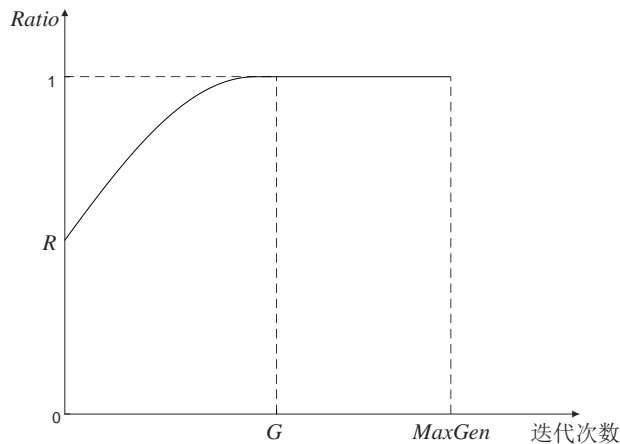


图 4-4 选择比例和迭代次数的关系曲线

$$Ratio = \begin{cases} R + (1-R) * \sin\left(\frac{G_c}{2G} \pi\right) & \text{if } 1 < G_c < G \\ 1 & \text{if } G \leq G_c \leq MaxGen \end{cases} \quad (4-11)$$

其中， $G_c$  为当前遗传代数。当迭代次数为 1 时，前级选择初始比率为  $R(0 < R < 1)$ ，当遗传代数为  $G(1 < G < MaxGen)$  时，选择比例为 1。

#### 4.1.4 IMSDE 算法描述

本节将给出改进算法的主要流程框架。算法具体步骤描述如下：首先，在初始化过程中对随机数进行 Tent 混沌映射和反向学习以生成更加均匀分布的随机数用于生成初

始种群；然后，使用算法 4.1 中的变异策略生成下一代种群。最终，根据自适应选择策略选择前  $N$  个个体进入下一代，重复上述过程至满足结束条件。算法主要流程的伪代码如算法 4.2 所示。

---

#### 算法 4.2 IMRSE 算法

---

输入：种群大小  $N$ 、最大迭代次数  $MaxGen$

输出：种群  $P_t$

- 1  $t \leftarrow 1$ ;
  - 2 对随机数进行 Tnet 混沌映射，初始化  $P_t = \{\mathbf{x}'_1, \dots, \mathbf{x}'_N\}$ ;
  - 3 对  $P_t$  进行反向学习得到反向种群  $P'_t$ ,  $P_t = \{P_t, P'_t\}$ ;
  - 4 从  $P_t$  中选择前  $N$  个最优个体为初始种群;
  - 5  $P_{t+1} \leftarrow \emptyset$ ;
  - 6 while  $t \leq MaxGen$  do
  - 7     根据种群  $P_t$  生成子代种群  $O_t$ ;
  - 8     对  $O_t$  中不可行个体进行合法化处理;
  - 9      $H_t \leftarrow P_t \cup O_t$ ;
  - 10    根据自适应选择策略从  $H_t$  中选择  $N$  个最优个体到  $P_{t+1}$ ;
  - 11     $t \leftarrow t+1$ ;
  - 12 end
- 

#### 4.1.5 IMSDE 算法复杂度分析

如算法 4.2 所示，IMSDE 算法的时间复杂度主要由选择策略及子代种群的生成决定。在最坏的情况下，计算 CDP 和拥挤距离的时间复杂度也为  $O(MN^2)$ 。因此，自适应选择策略的时间复杂度为  $O(MN^2)$ 。在子代种群的生成中，在最坏情况下，所有生成子代均不可行，时间复杂度为  $O(ND)$ ，如果需要在可行个体上使用 Pareto 优势选择最佳个体，则时间复杂度为  $O(MN^2)$ 。综上所述，IMSDE 算法的总体时间复杂度为  $O(MN^2)$ ，与大多数现有的 MOEA 相同，例如 NSGA-II，时间复杂度同样为  $O(MN^2)$ 。算法的空间复杂度主要取决于种群规模和拥挤距离矩阵的计算。种群的空间复杂度为  $O(N^2 + ND)$ ，同样与大多数现有的 MOEA 相同。其中  $M$ 、 $N$  和  $D$  分别目标数、种群规模以及决策向量维度。

#### 4.1.6 IMSDE 算法性能分析

算法参数设置：每个测试函数上的种群大小设置为 100。对于每个测试函数，每个算法都独立实现 30 次取结果的平均值和标准差，并在函数评价次数到达 100000 次时停止。在模拟二元交叉中，交叉概率和分布指数分别设置为 1 和 20。在多项式变异中，变异概率和分布指数分别为  $1/D$  和 20。对于 DE 算子，收缩因子  $F$  的初始值取值为 0.5，交叉概率  $CR$  为 1.0。所涉及的比较算法中的其他参数设置为与原始论文相同，以实现公

平比较。

**评价指标:** 为了更加全面的衡量多目标优化算法的性能, 实验使用了两个性能指标: 反向世代距离 (Inverted generational distance, IGD) 和超容量 (Hyper volume, HV)。较小的 IGD 值和较大的 HV 值代表着更好的收敛性和多样性性能。同时为了更好的比较不同算法间的性能, 本文在 5% 显著性水平上使用 Wilcoxon 符号秩检验来检验 IMSDE 与其他算法的显著性差异。检验结果在每个表的最后一行给出。‘+/-/=’表示所提出算法的效果明显优于/明显差于/基本相同当前算法。

**对比算法:** 为了验证改进后的 IMSDE 算法的有效性, 本节选取了 4 种 CMOEA 在 CF 测试函数上<sup>[80]</sup>进行系统的性能比较。4 个 CMOEA 分别是 C-MOEA/D<sup>[81]</sup>、AGEMOEAI<sup>[82]</sup>、NAGA-II-ToR<sup>[83]</sup>和 DSPCMDE<sup>[84]</sup>。由于本文所提出的 IMSDE 算法实在 GDE3<sup>[85]</sup>的基础上进行改进, 因此, 本章将 IMSDE 与 GDE3 作为消融实验进行对比。所有算法都在相同条件下执行, 相等条件意味着使用相同的起始和终止标准、相同数量的起始搜索点、相同的数据集、相同的硬件运行算法。为了减少随机性的影响, 所有实验均执行 30 次。

**测试函数:** DOC 测试套件<sup>[86]</sup>具有从现实世界应用中收集的各种决策约束 (例如不等式约束、等式约束、线性约束和非线性约束), 使得决策空间中的可行域具有不同性质 (例如非线性、极小和多峰)。同时, 设计了一些可控的、简单的目标约束来缩小目标空间中的可行域, 使 PF 具有连续、离散、混合等特性。其中 DOC1-DOC7 具有两个目标, DOC8、DOC9 具有三个目标。与其他人工 CMOPs 相比, DOC 具有更好的模拟真实场景的能力。CF 的 10 个基准测试函数的一些信息如表 4-1 所示。

表 4-1 CF 测试函数

函数	目标数	维度	搜索范围	PF 形状
CF1	2	10	$[0,1]^{10}$	离散
CF2	2	10	$[0,1] \times [-1,1]^9$	不连通
CF3	2	10	$[0,1] \times [-2,2]^9$	不连通
CF4	2	10	$[0,1] \times [-2,2]^9$	线性
CF5	2	10	$[0,1] \times [-2,2]^9$	线性
CF6	2	10	$[0,1] \times [-2,2]^9$	线性、凹
CF7	2	10	$[0,1] \times [-2,2]^9$	线性、凹
CF8	3	10	$[0,1]^2 \times [-4,4]^8$	不连通
CF9	3	10	$[0,1]^2 \times [-2,2]^8$	曲面
CF10	3	10	$[0,1]^2 \times [-2,2]^8$	曲面

表 4-2 和表 4-3 分别给出了算法在 CF 数据集上的 IGD 和 HV 的对比结果。在 CF 数据集上通过测试各函数可以看出: 在 CF 函数集上使用 IMSDE 算法明显优于其他算法, 以 CF5 为例, 采用 IMSDE 算法时 IGD 和 HV 指标的值分别为 1.8331E-1、3.4304E-1, 与其余算法中最优相比, IGD 降低了约 21.5%, HV 增高了约 9.21%。



表 4-2 对比算法在测试函数上的 IGD 指标对比

Problem	Statistics	NSGA-II-ToR	C-MOEA/D	AGEMOEAI	DSPCMDE	IMSDE
CF1	Mean	2.0423e-2	3.3648e-2	2.0977e-2	9.8410e-3	<b>7.4186e-3</b>
	(Std Dev)	(2.30e-3) -	(4.29e-3) -	(2.56e-3) -	(3.66e-3) -	<b>(1.28e-3)</b>
CF2	Mean	4.9024e-2	1.1529e-1	5.2994e-2	2.3097e-2	<b>1.5404e-2</b>
	(Std Dev)	(2.36e-2) -	(4.48e-2) -	(1.92e-2) -	(1.15e-2) -	<b>(9.81e-3)</b>
CF3	Mean	2.4618e-1	2.9868e-1	2.7866e-1	1.4121e-1	<b>8.0701e-2</b>
	(Std Dev)	(7.47e-2) -	(9.36e-2) -	(1.16e-1) -	(7.26e-2) -	<b>(1.09e-1)</b>
CF4	Mean	1.0588e-1	1.5036e-1	1.0775e-1	6.7259e-2	<b>4.9174e-2</b>
	(Std Dev)	(3.72e-2) -	(6.22e-2) -	(3.65e-2) -	(6.26e-3) -	<b>(1.05e-2)</b>
CF5	Mean	3.4712e-1	3.6840e-1	2.9051e-1	2.3354e-1	<b>1.8331e-1</b>
	(Std Dev)	(1.33e-1) -	(1.32e-1) -	(1.03e-1) -	(1.62e-1) -	<b>(1.12e-1)</b>
CF6	Mean	8.8963e-2	1.7406e-1	9.0769e-2	4.4665e-2	<b>3.7697e-2</b>
	(Std Dev)	(3.25e-2) -	(5.14e-2) -	(3.40e-2) -	(1.49e-2) -	<b>(1.21e-2)</b>
CF7	Mean	3.1402e-1	3.1550e-1	3.6029e-1	1.3974e-1	<b>1.2273e-1</b>
	(Std Dev)	(1.14e-1) -	(1.08e-1) -	(1.62e-1) -	(4.76e-2) =	<b>(5.70e-2)</b>
CF8	Mean	4.3450e-1	3.2918e-1	1.8796e-1	<b>1.8615e-1</b>	3.6407e-1
	(Std Dev)	(4.63e-2) -	(1.01e-1) =	(6.27e-2) +	<b>(1.68e-2) +</b>	(8.97e-2)
CF9	Mean	1.5391e-1	1.2670e-1	1.6570e-1	1.1491e-1	<b>9.2406e-2</b>
	(Std Dev)	(2.59e-2) -	(1.97e-2) -	(6.95e-2) -	(8.38e-3) -	<b>(7.74e-3)</b>
CF10	Mean	NaN	NaN	NaN	4.2431e-1	<b>4.0615e-1</b>
	(Std Dev)	(NaN)	(NaN)	(NaN)	(1.08e-1) =	<b>(1.60e-1)</b>
DOC1	Mean	2.6181e+0	3.8598e+0	2.3216e+0	4.7335e+2	<b>7.3028e-3</b>
	(Std Dev)	(1.77e+0) -	(4.16e+0) -	(1.59e+0) -	(4.60e+2) -	<b>(4.30e-4)</b>
DOC2	Mean	NaN	NaN	NaN	4.4258e-1	<b>4.4112e-1</b>
	(Std Dev)	(NaN)	(NaN)	(NaN)	(4.57e-3) -	<b>(1.51e-4)</b>
DOC3	Mean	5.7733e+2	5.9637e+2	6.6065e+2	<b>1.1475e+2</b>	3.3263e+2
	(Std Dev)	(2.17e+2) -	(1.59e+2) -	(1.56e+2) -	<b>(2.70e+2) +</b>	(1.51e+2)
DOC4	Mean	7.6097e-1	6.9550e-1	1.2100e+0	4.9045e-2	<b>1.6108e-2</b>
	(Std Dev)	(8.22e-1) -	(4.40e-1) -	(1.94e+0) -	(1.24e-1) -	<b>(1.38e-3)</b>
DOC5	Mean	NaN	NaN	NaN	NaN	<b>8.7205e+0</b>
	(Std Dev)	(NaN)	(NaN)	(NaN)	(NaN)	<b>(3.31e+1)</b>
DOC6	Mean	2.9682e+0	1.3207e+0	1.8269e+0	2.9548e-3	<b>2.6336e-3</b>
	(Std Dev)	(3.72e+0) -	(9.36e-1) -	(2.30e+0) -	(1.71e-4) -	<b>(8.07e-5)</b>
DOC7	Mean	6.2441e+0	6.1305e+0	5.4160e+0	7.4962e-2	<b>2.3949e-3</b>
	(Std Dev)	(1.96e+0) -	(2.21e+0) -	(2.57e+0) -	(2.16e-1) -	<b>(8.24e-5)</b>
DOC8	Mean	6.2345e+1	8.3918e+1	8.4764e+1	<b>1.0813e-1</b>	1.2175e-1
	(Std Dev)	(5.43e+1) -	(6.03e+1) -	(7.58e+1) -	<b>(5.72e-3) +</b>	(8.35e-3)
DOC9	Mean	2.0068e-1	4.1623e-1	2.3075e-1	<b>1.1216e-1</b>	1.1858e-1
	(Std Dev)	(9.37e-2) -	(5.91e-2) -	(1.47e-1) -	<b>(1.14e-2) =</b>	(1.44e-2)
+/-/=		0/16/0	0/15/1	1/15/0	3/12/3	/

表 4-3 对比算法在测试函数上的 HV 指标对比

Problem	Statistics	NSGA-II-ToR	C-MOEA/D	AGEMOEAII	DSPCMDE	IMSDE	
CF1	Mean	5.4041e-1	5.2419e-1	5.3988e-1	5.5453e-1	<b>5.5707e-1</b>	
	(Std Dev)	(2.99e-3) -	(5.38e-3) -	(3.29e-3) -	(4.40e-3) -	<b>(1.55e-3)</b>	
CF2	Mean	6.0850e-1	5.5121e-1	6.0345e-1	6.5007e-1	<b>6.6128e-1</b>	
	(Std Dev)	(2.95e-2) -	(3.10e-2) -	(3.06e-2) -	(1.18e-2) -	<b>(6.56e-3)</b>	
CF3	Mean	1.7687e-1	1.5218e-1	1.6757e-1	2.4175e-1	<b>2.7870e-1</b>	
	(Std Dev)	(3.83e-2) -	(4.09e-2) -	(4.47e-2) -	(6.74e-2) -	<b>(7.67e-2)</b>	
CF4	Mean	4.1268e-1	3.7065e-1	4.0921e-1	4.4908e-1	<b>4.7127e-1</b>	
	(Std Dev)	(2.32e-2) -	(4.41e-2) -	(2.77e-2) -	(6.20e-3) -	<b>(7.84e-3)</b>	
CF5	Mean	2.5396e-1	2.4783e-1	2.8892e-1	3.1410e-1	<b>3.4304e-1</b>	
	(Std Dev)	(6.88e-2) -	(6.13e-2) -	(6.10e-2) -	(9.37e-2) -	<b>(7.64e-2)</b>	
CF6	Mean	6.3416e-1	5.9621e-1	6.2780e-1	6.5979e-1	<b>6.6324e-1</b>	
	(Std Dev)	(1.46e-2) -	(3.38e-2) -	(2.10e-2) -	(8.13e-3) =	<b>(7.58e-3)</b>	
CF7	Mean	4.2413e-1	4.3653e-1	3.9985e-1	5.3126e-1	<b>5.4834e-1</b>	
	(Std Dev)	(7.79e-2) -	(6.59e-2) -	(1.14e-1) -	(4.71e-2) =	<b>(4.14e-2)</b>	
CF8	Mean	1.6910e-1	2.4966e-1	<b>3.4189e-1</b>	3.2883e-1	2.4856e-1	
	(Std Dev)	(2.18e-2) -	(2.27e-2) =	<b>(3.65e-2) +</b>	(2.12e-2) +	(3.75e-2)	
CF9	Mean	3.3786e-1	3.6832e-1	3.8048e-1	4.0568e-1	<b>6.4554e-1</b>	
	(Std Dev)	(2.81e-2) -	(3.55e-2) -	(4.57e-2) -	(1.14e-2) -	<b>(1.04e-2)</b>	
CF10	Mean	NaN	NaN	NaN	1.0423e-1	<b>1.3237e-1</b>	
	(Std Dev)	(NaN)	(NaN)	(NaN)	(5.44e-2) =	<b>(6.80e-2)</b>	
DOC1	Mean	5.2028e-3	1.1258e-2	1.9352e-2	1.6115e-1	<b>3.4103e-1</b>	
	(Std Dev)	(2.52e-2) -	(3.01e-2) -	(5.69e-2) -	(1.76e-1) =	<b>(1.18e-3)</b>	
DOC2	Mean	NaN	NaN	NaN	3.3678e-1	<b>3.4114e-1</b>	
	(Std Dev)	(NaN)	(NaN)	(NaN)	(1.21e-2) -	<b>(1.73e-4)</b>	
DOC3	Mean	0.0000e+0	0.0000e+0	0.0000e+0	<b>2.1103e-1</b>	5.4004e-2	
	(Std Dev)	(0.00e+0) =	(0.00e+0) =	(0.00e+0) =	<b>(1.50e-1) +</b>	(1.23e-1)	
DOC4	Mean	1.5426e-1	7.4525e-2	1.1020e-1	5.1771e-1	<b>5.4485e-1</b>	
	(Std Dev)	(1.67e-1) -	(7.66e-2) -	(1.59e-1) -	(8.11e-2) -	<b>(1.76e-3)</b>	
DOC5	Mean	NaN	NaN	NaN	NaN	<b>4.5775e-1</b>	
	(Std Dev)	(NaN)	(NaN)	(NaN)	(NaN)	<b>(1.24e-1)</b>	
DOC6	Mean	4.4314e-2	5.4807e-2	7.7675e-2	5.2661e-1	<b>5.3484e-1</b>	
	(Std Dev)	(1.03e-1) -	(1.13e-1) -	(1.49e-1) -	(6.82e-3) -	<b>(4.23e-3)</b>	
DOC7	Mean	0.0000e+0	0.0000e+0	0.0000e+0	4.5474e-1	<b>5.3481e-1</b>	
	(Std Dev)	(0.00e+0) -	(0.00e+0) -	(0.00e+0) -	(1.50e-1) -	<b>(7.27e-3)</b>	
DOC8	Mean	0.0000e+0	0.0000e+0	4.0348e-4	<b>7.3562e-1</b>	7.0817e-1	
	(Std Dev)	(0.00e+0) -	(0.00e+0) -	(2.21e-3) -	<b>(6.92e-3) +</b>	(1.15e-2)	
DOC9	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	
	(Std Dev)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)	
		+/-/=	0/14/1	0/13/2	1/13/1	3/10/4	/

如表 4-2 所示, IMSDE 算法在 15 个测试函数上 IGD 指标都取得了最优结果。相比于 DSPCMDE, IMSDE 在 CF8 表现不佳, 这是因为 CF8 的 PF 较为复杂, 而 DSPCMDE 利用了不可行解的信息可以帮助种群突破不可行区域的隔离搜索到更多的解。与 DSPCMDE 总体对比来说, IMSDE 有着明显的优势, 这是因为 DSPCMDE 在前期关注

了太多的不可行解信息,而对于可行解的信息随之减弱,导致不能更好的收敛,而 IMSDE 的种群在初始化时有着更好的分布,且在变异操作过程中提高了种群的多样性,这意味着可以更好的收敛。

Wilcoxon 符号秩检验结果也显示了 IMSDE 算法具有优异的性能表现,在指标 IGD 上分别在 16, 15, 15, 12 个测试函数上显著优于其他对比算法,在指标 HV 上分别在 14, 13, 13, 10 个测试函数上显著优于其他对比算法。如表 4-3 所示,IMSDE 算法在 15 个测试函数上取得了最优结果,AGEMOEAI 在 CF8 测试函数上优于 IMSDE 算法,DSPCMDE 在 CF8、DOC3、DOC8 三个测试函数上优于 IMSDE 算法,但是分别在 13 个和 10 个测试函数上显著差于 IMSDE。需要注意的是,IMSDE 在所有测试函数上均能获得 100%FR,而其他算法在 CF8、CF10 或者 DOC5 上由于 PF 被不可行区域断开和阻碍导致不能在每次运行中都提供可行的解决方案。

图 4-5 和图 4-6 分别给出各个算法在 CF2 函数上的收敛曲线和种群分布。从图 4-5 可以看出 IMSDE 算法取得了较好的收敛效果,在 30000 次评价次数时接近收敛。由图 4-6 可知,IMSDE 算法在 Pareto 前沿上分布相对于 NSGA-II-ToR、C-MOEA/D、AGEMOEAI 和 DSPCMDE 分布都更加均匀,且收敛性更好,DSPCMDE 分布不够均匀,且部分点未收敛。这是因为 IMSDE 采用的初始化方法和变异修补算子均提升了种群的多样性,改善了种群的分布情况,且变异策略中使用了两个变异算子及自适应选择策略均较好的平衡了种群的多样性和收敛性。

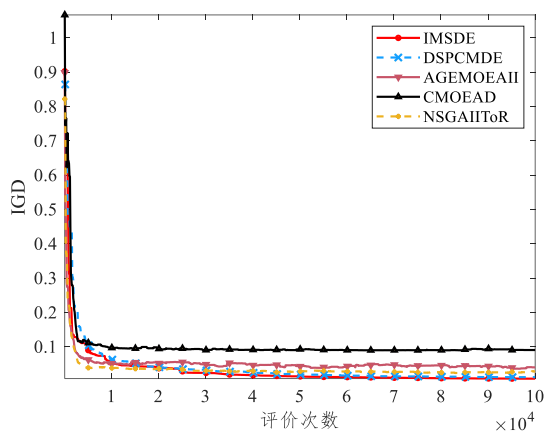


图 4-5 CF2 函数上的收敛曲线

消融实验:本章节提出了反向混沌初始化和变异策略,在初始化阶段,利用 Tent 混沌映射和反向学习扩大了搜索方向,提高了种群的多样性,改善了种群的分布情况。变异过程中,使用了两种流行 DE 算子 DE/rand-to-best/1/bin 和 DE/current-to-rand/1 进行收敛性和多样性的平衡,并对产生后代个体进行合法化处理使其满足解空间范围,进一步保证了进化过程中个体的多样性。在环境选择过程中,并不是所有处于最高等级的个体都被选中,因为其中一些个体可能非常拥挤,相反,底层的潜在解具有成为更优解的潜力,针对该问题设计的自适应选择策略能够保证种群的多样性和收敛性。为了验证改进策略的有效性,表 4-4、4-5 分别给出了包含不同策略的算法的 IGD 和 HV 结果对比,其中命名中包含 I 代表反向混沌初始化策略、M 代表变异策略及修补算子、S 代表自适

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/225113331003011114>