

考虑服务顺序和平衡性MOVRPTW问题的EASS-HHO算法

张天翼, 张惠珍*, 李留留

上海理工大学管理学院, 上海

收稿日期: 2025年7月29日; 录用日期: 2025年8月13日; 发布日期: 2025年9月30日

摘要

考虑服务顺序和平衡性带时间窗的多目标车辆路径问题, 以最小化配送成本, 最小化车辆流转时间差和最大化顾客满意度为目标, 构建模型。针对该模型特点, 设计了一种边际采样-哈里斯·鹰算法对该模型进行求解。最后使用所罗门算例测试, 结果表明: 边际采样-哈里斯·鹰算法求解结果比基本哈里斯·鹰算法得到了改善。该算法求解现有文献中实例也较文献中结果更优。求解性能与其它启发式算法相比较该算法性能更高效。

关键词

多目标, 平衡性, 服务顺序约束, 车辆路径问题, 哈里斯·鹰算法

EASS-HO Algorithm for MOVRPTW Problems with Service Order and Balance

Tianyi Zhang, Huizhen Zhang*, Liuliu Li

Business School, University of Shanghai for Science and Technology, Shanghai

Received: July 29, 2025; accepted: August 13, 2025; published: September 30, 2025

Abstract

A multi-objective vehicle routing problem with time windows considering service order and balance is constructed to minimize delivery costs, minimize vehicle flow time differences, and maximize customer satisfaction. A marginal sampling Harris Eagle algorithm was designed to solve the model based on its characteristics. Finally, a Solomon example was used for testing, and the results showed that the marginal sampling Harris Eagle algorithm improved the solution results compared

*通讯作者。

文章引用: 张天翼, 张惠珍, 李留留. 考虑服务顺序和平衡性 MOVRPTW 问题的 EASS-HHO 算法[J]. 运筹与模糊学, 2025, 15(5): 87-99. DOI: 10.12677/orf.2025.155234

to the basic Harris Eagle algorithm. This algorithm also performs better in solving examples in existing literature than in literature. Compared with other heuristic algorithms, this algorithm performs more efficiently in solving problems.

Keywords

Multi-Objective, Balance, Service Order Constraints, Vehicle Routing Problem, Harris Hawks Optimizer Algorithm

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着顾客所需服务的种类逐渐增多以及电子商务等行业的蓬勃发展,我国物流配送行业需要将顾客的多种需求考虑到路径规划中,以期达到物流公司与顾客的双赢。针对顾客有多种服务需求问题,国内外学者对此进行了相关研究。如:Heechul Bae 等[1]针对先交付后安装电子设备的配送要求,考虑时间窗,多站点车辆路径,最大运输时间等因素,构建了系统成本最小化的车辆路径问题模型。Ousmane Ali 等[2]针对先配送家具再安装的要求,使用异构车辆来进行服务。第一个车队负责交付,不可以安装产品,而第二个车队只执行安装,构建了车辆成本最小化的车辆路径问题模型。庞海军等[3]将产品的配送与安装的同步性进行分离,建立以最小化配送和安装时间为目标的车辆路径问题模型。

在现有文献中针对顾客有多种需求时,考虑有服务顺序配送问题,同时考虑物资配送人员工作强度平衡性的多目标车辆路径问题较少。因此,文章建立了考虑服务顺序和平衡性带时间窗的多目标车辆路径问题模型(Multi-Objective Vehicle Routing Problem with Time Windows and Service Order and Balance, MOV-RPTW-SOB)。

车辆路径问题属于 NP 难问题。小规模问题时,可以使用精确算法求解(如分枝定界法[4],列生成法[5]等)求解,但实际物流配送中抽象出来的车辆路径问题往往规模较大,此时,使用精确算法在可接受时间内得到可接受的解决方案较难[6]。故,在有限时间范围内得到可接受的解的智能优化算法[7]成为了解决车辆路径问题的主要方法。如:遗传算法[8],蚁群算法[9]和模拟退火算法[10]。

哈里斯·鹰优化算法[11](Harris Hawks Optimizer Algorithm, HHO)是一种新型智能优化算法,其已被成功应用于解决汽车工业中的形状优化问题[12],卫星图像分割[13]等多个领域,但极少应用于物流配送领域内的选址及车辆路径等离散型问题的求解。

本文使用 HHO 算法求解文章构建的多目标车辆路径问题模型,同时根据模型的具体特点与 HHO 算法本身特性,将边缘采样策略[14](Edge Area-Based Sampling Strategy, EASS)与 HHO 算法进行结合,设计求解该模型的边缘采样-哈里斯·鹰算法(Edge Area-Based Sampling Strategy-Harris Hawks Optimizer Algorithm, EASS-HHO)。在案例背景下,通过将 EASS-HHO 算法与基本哈里斯·鹰算法(Basic Harris Hawks Optimizer Algorithm, BHHO)的求解结果进行对比分析,以验证算法与模型的有效性。

2. 考虑服务顺序和平衡性带时间窗的多目标车辆路径问题模型

2.1. 问题描述

在考虑服务顺序和平衡性的多目标带时间窗车辆路径问题中,需要根据客户的地理分布,为其安排

由不同类型车辆执行的多项服务，并保证服务顺序得到遵循。模型的优化目标包括：在满足服务约束的同时，降低整体配送成本、缩小车辆间的流转时间差，并尽量提升客户满意度。其基本约束条件如下：

(1) 每类服务由对应类型的车辆承担，且每辆车的行驶时间不得超过其限定的最长时长；(2) 各客户对任一服务的需求量不能超出车辆可提供的最大容量，且每种服务需求需一次性完成；(3) 客户可能存在单项或多项服务需求，若存在多项，则须按照预先设定的服务顺序依次完成；(4) 所有车辆均从同一配送中心发车，完成路径中所有客户的服务后需回到起点；(5) 对于含时间窗约束的服务，若车辆在规定时间内到达，客户满意度记为 1；若超出时间窗，则满意度为 0，并产生随早到或延迟时间线性增加的惩罚值，惩罚系数记为常数 b 。

2.2. 数学模型

模型主要参数设置如下：

J ：配送网络顶点(顶点 0 为仓库，其它顶点为 n 名顾客)集合($j \in J, j = 0, 1, \dots, n$)； P ：服务需求种类集合($p \in P, p = 1, \dots, p$)； A^p ：表示对第 p 类服务有需求的客户节点组成的集合； H^p ：可执行第 p 类服务的车辆组成的集合； q_j^p ：客户 j 对第 p 类服务的具体需求数量； Q^p ：执行第 p 类服务的车辆所允许的最大服务容量； T^p ：该类车辆单次任务中允许的最长行驶时间； sl ：对于同一客户而言，其接受多种服务时，服务之间最大允许的时间间隔； S_{jv}^p ：车辆 v 为顾客 j 提供第 p 类服务的服务时间； d_{ij} ：顾客 i 与顾客 j 之间的距离； C^p ：提供第 p 类服务的车辆的启用费； G^p ：提供第 p 类服务的车辆的单位行驶距离费； a_{jv}^p ：表示车辆 v 在为客户 j 执行第 p 类服务时的到达时间； t_{jv}^p ：表示车辆 v 执行第 p 类服务时，从客户 i 前往客户 j 所需的行驶时间； $[T_j^{lp}, T_j^{hp}]$ ：客户 j 在接受第 p 类服务时所设定的时间窗区间。

决策变量如下：

$$x_{jv}^p = \begin{cases} 1, & \text{提供 } p \text{ 类服务的车辆 } v \text{ 从顾客 } i \text{ 直接行驶到顾客 } j, \forall p \in P, i, j \in J, v \in H^p; \\ 0, & \text{否则} \end{cases}$$

$$y_{jv}^p = \begin{cases} 1, & \text{提供 } p \text{ 类服务的车辆 } v \text{ 为顾客 } j \text{ 提供服务}, \forall p \in P, j \in J, v \in H^p; \\ 0, & \text{否则} \end{cases}$$

$$z_v^p = \begin{cases} 1, & \text{提供 } p \text{ 类服务的车辆 } v \text{ 被使用}, \forall p \in P, v \in H^p; \\ 0, & \text{否则} \end{cases}$$

e_{jv}^p ：车辆 v 到达客户 j 提供第 p 类服务时所产生的提前等待时间； l_{jv}^p ：当车辆 v 为客户 j 提供第 p 类服务但未能按时到达时，其产生的延迟时间； CS_j^p ：客户 j 对第 p 类服务的满意度水平； F_j^p ：顾客 j 在接受第 p 类服务时，由于车辆早到与晚到产生的惩罚值； TT_v^p ：提供第 p 类服务的车辆 v 的总流转时间； h_v^p ：提供第 p 类服务的车辆 v 经过的顾客集合。

在上述参数和决策变量设置的基础上，考虑平衡性和服务顺序限制带时间窗的多目标车辆路径数学模型可构建如下：

$$\min f_1 = \sum_{p \in P} \sum_{j \in J} \sum_{v \in H^p} C^p x_{0jv}^p + \sum_{p \in P} \sum_{v \in H^p} \sum_{i \in J \cup A^p} \sum_{j \in J \cup A^p} G^p t_{ijv}^p x_{ijv}^p + \sum_{p \in P} \sum_{j \in J} F_j^p \quad (1)$$

$$\max f_2 = \sum_{p \in P} \sum_{j \in J} CS_j^p \quad (2)$$

$$\min f_3 = \min \sum_{p \in P} \left\{ \max \{ TT_v^p \mid v \in H^p \} - \min \{ TT_v^p \mid v \in H^p \} \right\} \quad (3)$$

s.t.

$$\sum_{i \in J \cap A^p} x_{jv}^p = y_{jv}^p, \forall j \in A^p, v \in H^p, p \in P \quad (4)$$

$$\sum_{i \in J \cap A^p} x_{jiv}^p = y_{jiv}^p, \forall j \in A^p, v \in H^p, p \in P \quad (5)$$

$$\sum_{v \in H^p} y_{jv}^p = 1, \forall j \in A^p, p \in P \quad (6)$$

$$\sum_{j \in A^p} y_{jv}^p q_j^p \leq Q^p, \forall v \in H^p, p \in P \quad (7)$$

$$a_{iv}^p + s_{iv}^p + t_{ijv}^p \leq a_{jv}^p + M(1 - x_{ijv}^p), \forall i, j \in J \cup A^p, v \in H^p, p \in P \quad (8)$$

$$a_{iv}^p + s_{iv}^p - M(1 - y_{iv}^p) \leq a_{iw}^{p+1} + M(1 - y_{iw}^{p+1}), \forall i \in A^p \cap A^{p+1}, v \in H^p, w \in H^{p+1}, p \in P \quad (9)$$

$$a_{iw}^{p+1} - M(1 - y_{iw}^{p+1}) \leq a_{iv}^p + s_{iv}^p + M(1 - y_{iv}^p) + sl, \forall i \in A^p \cap A^{p+1}, v \in H^p, w \in H^{p+1}, p \in P \quad (10)$$

$$\sum_{j \in J \cup A^p} \sum_{i \in J \cup A^p} x_{ijv}^p t_{ijv}^p \leq T^p, \forall v \in H^p, p \in P \quad (11)$$

$$T_j^{Ep} - M(1 - y_{jv}^p) \leq a_{jv}^p + e_{jv}^p - l_{jv}^p \leq T_j^{Lp} + M(1 - y_{jv}^p), \forall j \in A^p, v \in H^p, p \in P \quad (12)$$

$$CS_j^p = \begin{cases} 0, a_{jv}^p < T_j^{Ep}, \forall p \in P, j \in J \\ 0, T_j^{Lp} < a_{jv}^p, \forall p \in P, j \in J \\ 1, T_j^{Ep} \leq a_{jv}^p \leq T_j^{Lp}, \forall p \in P, j \in J \end{cases} \quad (13)$$

$$F_j^p = \begin{cases} b(T_j^{Ep} - a_{jv}^p), a_{jv}^p \leq T_j^{Ep}, \forall p \in P, j \in J \\ b(a_{jv}^p - T_j^{Lp}), T_j^{Lp} \leq a_{jv}^p, \forall p \in P, j \in J \end{cases} \quad (14)$$

$$TT_v^p = \sum_{i \in J \cup h_k^p} \sum_{j \in J \cup h_k^p} t_{ijv}^p + \sum_{j \in J \cup h_k^p} e_{jv}^p + \sum_{j \in J \cup h_k^p} s_{jv}^p, \forall v \in H^p, p \in P \quad (15)$$

$$a_{iv}^p \geq 0, \forall i \in A^p, v \in H^p, p \in P \quad (16)$$

$$z_v^p \in \{0, 1\}, \forall v \in H^p, p \in P \quad (17)$$

$$y_{jv}^p \in \{0, 1\}, \forall j \in A^p, v \in H^p, p \in P \quad (18)$$

$$x_{ijv}^p \in \{0, 1\}, \forall i, j \in A^p, v \in H^p, p \in P \quad (19)$$

目标函数(1)表示在满足所有约束的前提下, 最小化整体配送成本, 成本包括各类车辆的启动车费、行驶距离费用, 以及因车辆提前或延迟到达所产生的惩罚费用; 目标函数(2)是最大化所有客户的综合满意度; 目标函数(3)进一步优化车辆运行调度, 尽可能缩小各车辆间的流转时间差。约束(4)~(5)表示车辆服务约束表示若车辆 v 执行客户 j 的第 p 类服务, 则必须完成到达、服务并离开该客户的流程; 约束(6)表示单次满足约束: 客户 j 的每类服务均需由相应类型车辆一次性完成; 约束(7)是容量约束, 同类型车辆在服务的客户总需求量不得超过该车辆的最大承载量; 约束(8)是服务先后约束: 若客户 i 与 j 均有第 p 类服务需求且由同一车辆承担, 则到达客户 j 的时间需不早于到达客户 i 的时间与其服务时间及两点行驶时间之和; 约束(9)表示同一客户若需要两类服务, 第 $p+1$ 类车辆的到达时间不得早于第 p 类服务完成时间; 约束(10)是多服务时间间隔约束, 同一客户两类服务车辆到达时间间隔不得超过规定的最大间隔; 约束(11)表示每辆车累计行驶时间不得超过其允许的最大值; 约束(12)是时间窗约束, 车辆 v 到达客户 j 执行第 p 类服务的时间必须与该客户设定的时间窗匹配; 约束(13)为客户 j 在接受第 p 类服务时产生相应满意度; 约束(14)是指车辆在服务客户 j 的第 p 类需求时, 若早到或晚到, 将计算相应惩罚值; 约束(15)表

示车辆流转时间定义,用于衡量各车辆调度的平衡性;约束(16)表示车辆运行时间定义,用于确保调度可行;(17)~(19)变量取值约束,保证模型逻辑的可行性。

3. 求解 MOVRPTW-SOB 的边际采样 - 哈里斯·鹰算法

3.1. BHHO 算法

HHO (Harris Hawks Optimization)算法是一种新型群体智能优化方法。在该算法中,每只“鹰”代表一个搜索代理,其位置视为候选解;而在所有候选解中,表现最优的个体被视为当前猎物位置或最可能接近猎物的点。每一代中,个体会根据自身、猎物及其邻近成员的位置动态调整策略,以更新其下一轮的位置。

算法通过模拟鹰群在不同逃逸能量状态(即猎物逃跑能力)与随机扰动下的围捕行为,选择多种包围与攻击策略,实现搜索空间的局部与全局探索之间的动态平衡。整个优化过程可划分为两个核心阶段:勘测阶段(Exploration)用于大范围搜索潜在解,围捕阶段(Exploitation)则用于围绕猎物进行精细优化与逼近。

根据猎物的逃脱能量 E ,选择不同的捕猎包围策略,其值在每次迭代时在 $(-1,1)$ 之间随机产生, E 为猎物的当前逃脱能量。当 $|E| \geq 1$,捕食者在不同的捕食者邻域中搜索猎物(勘测阶段);当 $|E| < 1$ 时,捕食者在不同捕食者与预测猎物位置的邻域中搜索猎物(开发阶段);同时根据逃脱能量值和随机数的不同取值,进行不同的围捕行为寻找猎物。该算法具体内容参考文献[11]。

3.2. 边际采样 - 哈里斯·鹰算法

为了很好地求解文章 MOVRPTW-SOB 问题,本文首先对初始种群进行非支配排序,得到非支配父代解;再使用边缘采样策略和重新生成解个体相结合的方法对非支配父代解,进行更新。此外,根据文章模型的具体特点设计 4 种不同算子进行领域搜索,更新当前种群的支配解,以期搜索到更优的解。

3.2.1. 初始化

采用自然数编码方式:假定顾客数为 n , n 名顾客的编号分别为 $1,2,3,\dots,n$;编号 $n+1$ 区别不同服务,编号 0 区别不同车辆路径;用符号 & 表示顾客对当前种类服务没有需求,有 p 种服务时,每个解个体长度为 $p \times n + p - 1$ 。如有 2 种服务,5 名顾客的个体,其中 1~5 号顾客对第一种服务有需求且随机生成顾客序列为 [3 5 4 2 1],生成路径为 [3 5 4 2 1 0];3~5 号顾客对第二种服务有需求且随机生成顾客序列为 [4 5 3],生成路径为 [4 5 3 0],则此个体的可行解如图 1 所示。

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 5 | 4 | 2 | 1 | 0 | & | & | 4 | 5 | 3 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|

Figure 1. Representation of the solution

图 1. 解的表示

3.2.2. 边际采样策略

考虑到多目标问题解的分布均衡性,将边际采样策略和 EASS-HHO 算法结合起来,使得求解的解分布更加均匀。边际采样策略是指将各个单目标的最优解纳入父代解的组成中,一方面使得父代解的产生更加多样,另一方面也为探索更多更优后代解的可能性增大。边际采样策略的示例如图 2 所示。

3.2.3. 父代解的生成

父代解的生成主要是通过多种方法生成父代的方式,达到后代解的多样性增加的目的。父代解生成的具体过程可被描述如下:

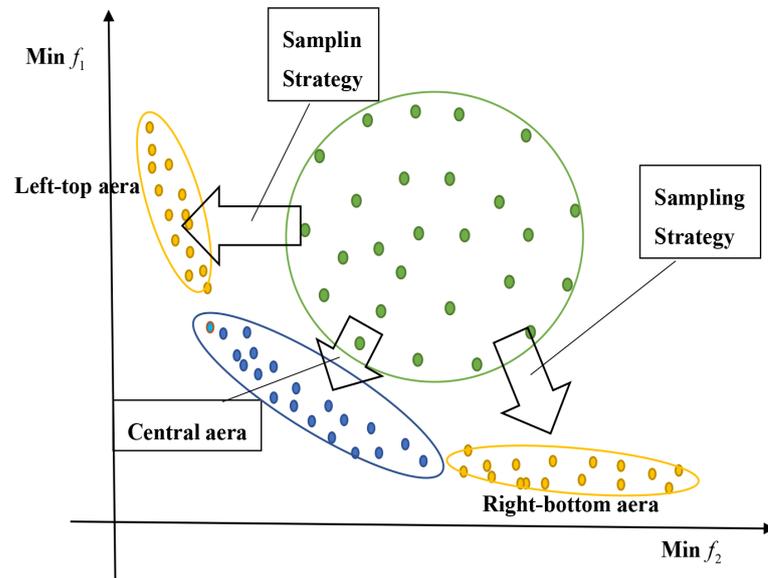


Figure 2. Marginal sampling
图 2. 边际采样

- 将每次迭代后的新种群进行非支配排序，得 n_1 个非支配解和 $N - n_1$ 支配解。
- 引入随机变量 r_1 , r_2 和变量 $JJ = t/T$, 比较随机变量 r_1 与变量 JJ 的大小, 当 $JJ > r_1$ 时, 转步骤 c); 否则, 非支配解为父代解。
- 使用随机变量 r_2 , 当 $r_2 < 0.5$ 时, 先利用边际采样策略生成 $n_1/2$ 个解, 用以随机取代 n_1 个非支配解中的 $n_1/2$ 个解, 然后与未取代的 $n_1/2$ 个解, 组成 n_1 个解为非支配父代解; 否则转步骤 d), 重新生成非支配解为父代解。
- 当支配解个数大于非支配解个数时, 从支配解中随机选取与非支配解个数相同的解为非支配解; 否则, 将所有支配解随机取代非支配解的部分解为部分非支配解。然后将非支配解中所有解的顾客访问序列依次取出, 然后按生成初始解的方法生成新的解, 进而将新的解作为非支配父代解。

3.2.4. 支配解的更新

为了进一步提高 EASS-HHO 算法在离散优化问题中的寻优能力, 本文在 BHHO 算法的基础上, 结合 MOVRPTW-SOB 问题的特点, 选择了组合算子和 4 种交叉算子更新支配解。

组合算子: 将生成的父代解, 先按各服务种类随机组合成新的多服务需求解, 然后和原有的父代解, 共同构成多个新解, 最后随机取其中的一个非支配解为子代解。假定有 n_1 个服务种类为 p 的父代解, 依次取 n_1 中第 p 种服务与其它 $n_1 - 1$ 个有 $p - 1$ 种服务的父代解组合为新的多服务需求解, 最终生成 n_1^p 个多服务需求解。如图 3 所示, 有 2 种服务类型的非支配父代解 Parent 1 [3 2 4 5 1 0 6 4 5 3 0 6] 与 Parent 2 [5 1 4 3 2 0 6 5 4 3 0 6], 进行组合操作算子得出后代解 Child 1, Child 2, Child 3 与 Child 4。

交叉算子 1 (次序交叉算子): 先将父代解两两分为一组(若有奇数父代解时, 随机选中一个父代解舍弃); 然后将两两父代解根据服务类型依次进行次序交叉, 形成新的解; 最后随机选取其中的一个非支配解为子代解。以图 1 当中的第一种服务为例, 假定有 2 个非支配父代解 Parent 1 [3 2 4 5 1 0 6] 和 Parent 2 [5 1 4 3 2 0 6], 随机选中进行次序交叉的顾客序列起止点分别为第 2 和第 4 个位置。对 Parent 1 而言, 顾客序列 [2 4 5] 保持不变, 剩余顾客相对访问序列 [3 1]; 同时, 用删除 Parent 2 上顾客 2, 4, 5 后剩余顾客相对访问序列 [1 3] 替换 Parent 1 剩余顾客相对访问序列 [3 1], 生成 Child 1 [1 2 4 5 3 0 6]; 同理, 对 Parent

2 也进行相应操作；最终得后代解 Child 1 和 Child 2，见图 4 所示。

| | | | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|---|---|
| Parent 1 | 3 | 2 | 4 | 5 | 1 | 0 | 6 | 4 | 5 | 3 | 0 | 6 |
| Parent 2 | 5 | 1 | 4 | 3 | 2 | 0 | 6 | 5 | 4 | 3 | 0 | 6 |
| Child 1 | 3 | 2 | 4 | 5 | 1 | 0 | 6 | 4 | 5 | 3 | 0 | 6 |
| Child 2 | 3 | 2 | 4 | 5 | 1 | 0 | 6 | 5 | 4 | 3 | 0 | 6 |
| Child 3 | 5 | 1 | 4 | 3 | 2 | 0 | 6 | 4 | 5 | 3 | 0 | 6 |
| Child 4 | 5 | 1 | 4 | 3 | 2 | 0 | 6 | 5 | 4 | 3 | 0 | 6 |

Figure 3. Operational procedure of the composite operator

图 3. 组合算子的操作流程

| | | | | | | | |
|----------|---|---|---|---|---|---|---|
| Parent 1 | 3 | 2 | 4 | 5 | 1 | 0 | 6 |
| Parent 2 | 5 | 1 | 4 | 3 | 2 | 0 | 6 |
| Child 1 | 1 | 2 | 4 | 5 | 3 | 0 | 6 |
| Child 2 | 2 | 1 | 4 | 3 | 5 | 0 | 6 |

Figure 4. Operation procedure of crossover operator 1 (Order Crossover)

图 4. 交叉算子 1 (次序交叉) 的操作流程

交叉算子 2: 该交叉操作以“顾客数最多”为评估依据, 从非支配父代解中随机选择一条包含客户最多的路径作为子代解的首条路径。选中的客户将被从所有父代解中移除, 并对相关路径数据进行同步更新。此过程不断重复, 直到所有客户均被成功分配到子代解中。以图 1 当中的第一种服务为例, 假定有 2 个非支配父代解 Parent 1 [3 2 0 4 5 1 0 6] 和 Parent 2 [5 1 4 3 0 2 0 6], 2 个父代中顾客数最多的路径 [5 1 4 3 0] 即为后代 Parent 2 的第一条路径, 剩余顾客 2 插入路径 [5 1 4 3 0] 的合法位置即生成了后代解 Child 1, 见图 5 所示。

| | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|
| Parent 1 | 3 | 2 | 0 | 4 | 5 | 1 | 0 | 6 |
| Parent 2 | 5 | 1 | 4 | 3 | 0 | 2 | 0 | 6 |
| Child 1 | 5 | 1 | 2 | 4 | 3 | 0 | 6 | |

Figure 5. Operational procedure of crossover operator 2

图 5. 交叉算子 2 的操作流程

交叉算子 3: 交叉算子 3 采用“单位客户路径距离最小”作为选择依据。在非支配父代解中, 随机选出一条其总行驶距离与客户数量之比最小的路径作为子代解的起始路径。被纳入该路径的客户将从所有父代解中剔除, 且相应路径数据进行更新。上述步骤持续执行, 直到所有客户均被覆盖并构建完成子代解。图 1 中展示的第一类服务可作为该算子应用的示例, 假定有 2 个非支配父代解 Parent 1 [3 2 0 4 5 1 0 6]

6]和 Parent 2 [5 1 4 3 0 2 0 6], 2 个父代解中行驶路径总距离与顾客总数的比值最小的路径[3 2 0]作为子代的第一条新路径, 剩余顾客 4, 5, 1 分别依次插入路径[3 2 0]的合法位置, 生成后代解 Child 1 [3 4 5 0 1 2 0 6], 见图 6 所示。

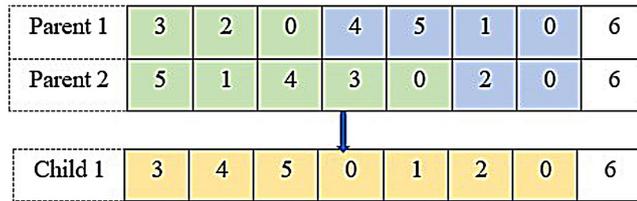


Figure 6. Operational procedure of crossover operator 3
图 6. 交叉算子 3 的操作流程

交叉算子 4: 该交叉策略首先在非支配父代解中, 随机选出一条顾客总等待时间与顾客数量比例最小的路径, 并将其作为子代解的起始路径。随后, 将该路径中涉及的顾客从所有父代解中移除, 并相应更新各父代路径中的相关信息。上述过程不断迭代, 直至所有顾客均被有效纳入子代路径中。图 1 中所示的第一类服务可作为该交叉算子的应用示例。以图 1 当中的第一种服务为例, 假定有 2 个非支配父代解 Parent 1 [3 2 0 4 5 1 0 6]和 Parent 2 [5 1 4 3 0 2 0 6], 2 个父代解中顾客总等待时间与顾客总数的比值最小的路径[5 1 4 3 0]作为子代的第一条新路径, 剩余顾客 2 插入路径[5 1 4 3 0]的合法位置, 生成后代解 Child 1 [5 2 0 1 4 3 0 6], 见图 7 所示。

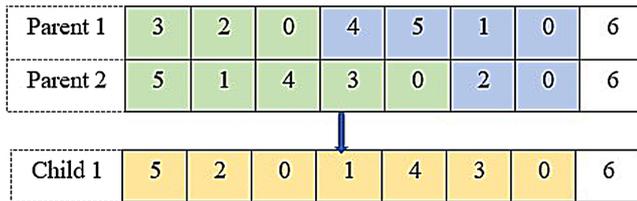


Figure 7. Operational procedure of crossover operator 4
图 7. 交叉算子 4 的操作流程

3.3. EASS-HHO 算法流程

根据前述算法步骤, 假定每代可得非支配解个数为 n_1 , EASS-HHO 算法的流程图如图 8 所示。

4. 算例分析

本文用 Windows10 搭建实验环境, 采用 Matlab2016a 对算法进行实现。为了更好地分析算法的优化性能, 文章使用 EASS-HHO 算法分别对 Solomon's Benchmark [15]算例库中的 c101~c105 算例的前 50 个顾客为有第一种服务需求且带时间窗[16]的顾客, 然后再随机选取其中的 20 个为有第二种服务需求且不带时间窗的顾客和 R207~R211 算例的前 100 个顾客为有第一种服务需求且带时间窗的顾客, 然后再随机选取其中的 40 个为有第二种服务需求且不带时间窗的顾客为实例和文献[17]当中的实例进行优化, 并对优化结果进行分析。

4.1. 参数设置和灵敏度分析

EASS-HHO 中种群大小 N 在很大程度上会影响其优化性能。为了确定最优 N 值和研究 N 对整体性能的影响, 分别取 Solomon's Benchmark 中 c101~c105 数据集和 R207~R211 数据集中数据, 对 EASS-HHO

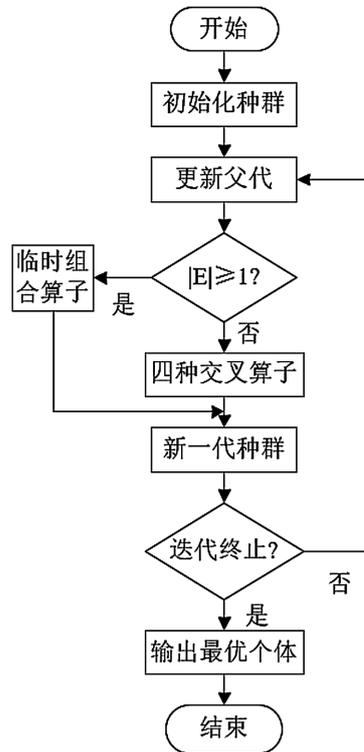


Figure 8. Algorithm flowchart
图 8. 算法流程图

的种群大小 N 进行灵敏度分析。

通过实验分析发现，当最大迭代次数 T 为 200 时算法整体优化性能较为理想。此外，为对非支配解集的质量进行客观评价，本文参考了 Ishibuchi [18] 等人提出的评估方法，对解集进行量化分析，其对应的评价指标如式(20)和式(21)所示。

$$R_{NDS}(S_j) = \left| S_j - \{x \in S_j \mid \exists y \in S : y \prec x\} \right| / |S_j| \quad (20)$$

$$NDS_NUM(S_j) = \left| S_j - \{x \in S_j \mid \exists y \in S : y \prec x\} \right| \quad (21)$$

其中， S 是不同种群规模得到的所有非支配解的集合； S_j 是种群规模为 N 时得到的非支配解的集合； $y \prec x$ 表示解 y 支配解 x ； $|S_j|$ 表示种群规模为 N 时得到的非支配解集的个数。指标 R_N 衡量解的质量，指标 N_N 衡量算法获得非支配解的能力，以 R_N 和 N_N 的均值为 $R_{N_{avg}}$ 和 $N_{N_{avg}}$ ，两个指标的值越大，表示此 N 值的效果越好。

表 1 中种群的大小分别被设置为 $0.5n$ 、 n 、 $1.5n$ 、 $2n$ 、 $2.5n$ 、 $3n$ (n 为实例中的顾客数)。由表 1 可知，当 $N=1.5n$ 时，指标 $R_{N_{avg}}$ 和指标 $N_{N_{avg}}$ 的值最大，算法性能最好，故在本文后续研究中种群数为 n 。

Table 1. Results for different dataset instances at different N values

表 1. 不同数据集实例在不同 N 下的运行结果

| N 值 | $0.5n$ | n | $1.5n$ | $2n$ | $2.5n$ | $3n$ |
|---------------|--------|-------|--------|-------|--------|-------|
| $R_{N_{avg}}$ | 0.150 | 0.133 | 0.213 | 0.169 | 0.173 | 0.151 |
| $N_{N_{avg}}$ | 1.200 | 1.600 | 3.800 | 2.600 | 3.400 | 2.600 |

4.2. 优化结果对比分析

利用 Solomon's Benchmark 中不同规模的独立数据集 C1 和 RC1 中的部分数据分别产生测试算例, 算例的数据设计参考文献[17]。为了更好地验证 EASS-HHO 算法中各个操作算子的优化性能, 在 EASS-HHO 算法其他参数保持不变的情况下, 分别将临时组合算子和 4 种交叉算子单独作为 EASS-HHO 算法操作算子, 运行 10 次, 找出所有非支配解中的最优成本, 最优满意度和最优车辆流转时间差, 结果如表 2~4 所示。

Table 2. Results of optimization with only the temporary composite operator and only crossover operator 1

表 2. 单独使用临时组合算子和单独交叉算子 1 的优化结果

| 算例 | 组合算子 | | | | | | 交叉算子 1 | | | | | |
|------|---------|-------|-------|---------|-------|--------|---------|-------|-------|---------|-------|--------|
| | 最优成本 | 最优满意度 | 最优时间差 | 最差成本 | 最差满意度 | 最差时间差 | 最优成本 | 最优满意度 | 最优时间差 | 最差成本 | 最差满意度 | 最差时间差 |
| C101 | 5087.7 | 44 | 178.7 | 6959.9 | 41 | 933.2 | 3292.9 | 45 | 101.5 | 7148.2 | 41 | 793.8 |
| C102 | 5205.3 | 44 | 232.5 | 7261.6 | 41 | 1120.1 | 3342.7 | 47 | 82.2 | 7334.1 | 41 | 829.5 |
| C103 | 4716.2 | 44 | 192.7 | 7458.5 | 41 | 1072.5 | 3445.4 | 46 | 61.3 | 6446.4 | 42 | 898.6 |
| C104 | 4682.2 | 44 | 127.2 | 7794.7 | 41 | 1168.4 | 3596.1 | 46 | 59.1 | 7101.1 | 41 | 1093.5 |
| C105 | 4836.8 | 43 | 185.6 | 8171.4 | 41 | 942.1 | 3584.8 | 46 | 90.6 | 6779.7 | 41 | 753.9 |
| R211 | 28081.5 | 54 | 487.4 | 35960.7 | 38 | 1543.7 | 25492.4 | 58 | 283.7 | 34856.5 | 40 | 1560.7 |

Table 3. Results of optimization with only crossover operator 2 and only crossover operator 3

表 3. 单独使用交叉算子 2 和单独交叉算子 3 的优化结果

| 算例 | 交叉算子 2 | | | | | | 交叉算子 3 | | | | | |
|------|---------|-------|-------|---------|-------|--------|---------|-------|-------|---------|-------|--------|
| | 最优成本 | 最优满意度 | 最优时间差 | 最差成本 | 最差满意度 | 最差时间差 | 最优成本 | 最优满意度 | 最优时间差 | 最差成本 | 最差满意度 | 最差时间差 |
| C101 | 1614.3 | 48 | 126.4 | 7099.3 | 41 | 1681.2 | 2209.1 | 49 | 103.4 | 7195.7 | 41 | 2122.8 |
| C102 | 1896.2 | 47 | 102.1 | 7362.8 | 41 | 1742.7 | 2303.1 | 48 | 105.6 | 7773.7 | 41 | 2617.1 |
| C103 | 1958.5 | 48 | 122.4 | 7728.7 | 41 | 1558.8 | 2007.6 | 48 | 114.1 | 7250.3 | 41 | 2157.9 |
| C104 | 1849.9 | 49 | 102.1 | 4235.6 | 44 | 2008.6 | 1912.1 | 48 | 108.8 | 8864.8 | 41 | 1993.9 |
| C105 | 1912.3 | 48 | 130.5 | 7032.1 | 41 | 1609.1 | 2198.9 | 47 | 122.1 | 7893.8 | 41 | 2446.9 |
| R211 | 10463.8 | 83 | 148.5 | 18422.4 | 69 | 5568.3 | 28076.1 | 55 | 605.4 | 35529.5 | 39 | 1733.2 |

Table 4. Results of optimization with only crossover operator 4

表 4. 单独使用交叉算子 4 的优化结果

| 算例 | 交叉算子 4 | | | | | |
|------|---------|-------|-------|---------|-------|--------|
| | 最优成本 | 最优满意度 | 最优时间差 | 最差成本 | 最差满意度 | 最差时间差 |
| C101 | 3831.9 | 44 | 190.9 | 7938.1 | 41 | 1904.9 |
| C102 | 4110.2 | 44 | 177.5 | 7046.6 | 41 | 1291.3 |
| C103 | 4282.1 | 43 | 177.8 | 7550.5 | 41 | 1738.7 |
| C104 | 4240.8 | 44 | 145.4 | 7841.1 | 41 | 1353.8 |
| C105 | 4768.5 | 44 | 187.1 | 7816.2 | 41 | 1145.9 |
| R211 | 26023.4 | 55 | 555.8 | 34053.5 | 40 | 1875.7 |

EASS-HHO 算法在数据集 r208, r209, c103, c105 上的多目标结果如表 5 所示。由表 5 可知, 当成本费较低时, 顾客满意度相对较小或车辆流转时间差相对较大; 当顾客满意度较大时, 成本费或车辆流转时间差相对较大; 当车辆流转时间差相对较小时, 成本费相对较大或顾客满意度相对较小。

Table 5. Performance of the EASS-HHO algorithm on four different datasets

表 5. EASS-HHO 算法在 4 种数据集上的优化结果

| 数据集 目标函数 | R208 | | | R209 | | | c103 | | | C105 | | |
|-------------|--------|-------|-------|--------|-------|-------|---------|-------|-------|---------|-------|-------|
| | f_1 | f_2 | f_3 | f_1 | f_2 | f_3 | f_1 | f_2 | f_3 | f_1 | f_2 | f_3 |
| 1 | 2161.2 | 45 | 81.2 | 2271.0 | 45 | 29.4 | 13424.0 | 23 | 753.5 | 16434.7 | 12 | 741.1 |
| 2 | 2161.2 | 45 | 90.4 | 2271.0 | 45 | 33.7 | 13424.0 | 23 | 715.6 | 16462.1 | 12 | 741.1 |
| 3 | 2349.5 | 46 | 47.0 | 2941.2 | 46 | 29.4 | 13424.0 | 23 | 721.5 | 16506.8 | 12 | 741.1 |
| 4 | 1913.1 | 46 | 103.8 | 2473.0 | 47 | 32.1 | 13424.0 | 23 | 753.5 | 16557.6 | 12 | 741.1 |
| 5 | 1917.3 | 46 | 103.8 | 2261.7 | 48 | 38.1 | 13424.0 | 23 | 775.5 | 15583.4 | 13 | 785.7 |
| 6 | 2372.2 | 46 | 56.2 | 1841.7 | 48 | 46.9 | 13470.1 | 23 | 715.6 | 17127.6 | 15 | 456.7 |
| 7 | 2382.4 | 46 | 56.2 | 1863.5 | 49 | 40.9 | 14621.0 | 25 | 463.1 | 17172.2 | 15 | 456.7 |
| 8 | 2386.5 | 46 | 56.2 | 1871.6 | 49 | 46.9 | 14667.0 | 25 | 463.1 | 16618.1 | 16 | 500.5 |
| 9 | 2388.4 | 46 | 50.2 | 1871.6 | 49 | 46.9 | 14210.1 | 25 | 494.9 | 10825.3 | 23 | 804.5 |
| 10 | 2394.0 | 46 | 50.2 | 1871.6 | 49 | 46.9 | 14089.5 | 26 | 506.2 | 10825.3 | 23 | 818.7 |
| 11 | 2398.1 | 46 | 50.2 | 1871.6 | 49 | 46.9 | 14089.5 | 26 | 516.8 | 11032.3 | 24 | 804.5 |
| 12 | 2323.3 | 46 | 57.6 | 1875.5 | 49 | 46.9 | 14135.6 | 26 | 506.2 | 10119.9 | 24 | 913.8 |
| 13 | 2003.1 | 47 | 762.5 | 1875.5 | 49 | 46.9 | 14778.0 | 26 | 453.9 | 10164.5 | 24 | 913.8 |
| 14 | 2297.5 | 48 | 679.1 | 1875.5 | 49 | 46.9 | 14778.0 | 26 | 489.8 | | | |
| 15 | 2300.4 | 48 | 679.1 | 1878.9 | 49 | 46.9 | 14824.1 | 26 | 453.9 | | | |
| 16 | 2303.1 | 48 | 679.1 | 1899.1 | 49 | 46.9 | | | | | | |

4.3. 算法性能比较

为评估所提出算法的性能, 本文采用两类指标: 收敛性指标 r 与解集分布性指标 Δ , 同文献[19]一样, 由于真实的最优帕累托前沿不可获得, 为便于评价, 我们将三种算法在所有实验中产生的非支配解集合的全体合并并去重, 作为一个近似最优帕累托解集, 用于与各算法结果进行比较分析。

收敛性指标: 用来评价当前所得解与真实最优解的逼近程度, 如式(22)所示, N 为最优 Pareto 解中解的数量, d_i 表示当前算法所得 Pareto 解集里的第 i 个解与最优 Pareto 解中的最近解的距离, 当 r 值越小, 表明解集的分布越均匀, 代表算法在维持多样性方面的表现越好。

$$r = \frac{\sum_{i=1}^N d_i}{N} \quad (22)$$

分布性指标: 用来评估当前算法所求得的 Pareto 解的均匀性, 如式(23)所示, d_j 和 d_l 分别为表示当前算法所得帕累托边界解与参考最优解集中两端极解之间的距离, \bar{d} 为表示参考最优解集中所有相邻解之间距离的平均值, d_i 为当前算法所得 Pareto 解集里第 i 个解与第 $i+1$ 间的距离, 当 Δ 值越小, 代表当

前算法均匀性越好。

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}} \quad (23)$$

同时, 为了方便计算, 文章使用欧式距离[20]代表两解之间的距离, 计算方法如式(24)所示, 其中 d_i 为第 i 个解与第 $i+1$ 个解之间的距离, $f_{j,i}$, $f_{j,i+1}$ 分别代表第 i 个解与第 $i+1$ 个解的第 j 个目标值, 由于文章是求解三个目标, 故 j 取值为 1、2、3。

$$d_i = \left(\sum_{j=1}^3 (f_{j,i} - f_{j,i+1})^2 \right)^{1/2} \quad (24)$$

利用前面所得数据计算各种算法的收敛性指标与分布性指标, 结果如表 6 所示。由表 6 可得, 从收敛性指标 r 来看, EASS-HHO 算法要优于 BHHO 算法及 NSGA-II, 从分布性指标 Δ 来看, EASS-HHO 算法与对比算法无显著差异, 略优于 BHHO 算法。

Table 6. Algorithm evaluation metrics

表 6. 算法评价指标

| 评价指标 | EASS-HHO | BHHO | NSGA-II |
|----------|----------|-------|---------|
| r | 20.45 | 24.43 | 22.67 |
| Δ | 0.47 | 0.48 | 0.47 |

5. 结论

本文围绕具备多类型服务需求且具有服务顺序约束的多目标车辆路径调度问题展开研究。问题特性包括: 客户具有时间窗约束, 多个服务需由不同车辆按顺序执行, 且车辆受限于最大服务容量和最远行驶时长。针对上述问题, 本文构建了一个以最小化配送成本、减少车辆间流转时间差和提升客户满意度为优化目标的多目标路径规划模型。

在算法设计方面, 结合问题特征与 HHO (Harris Hawks Optimization) 算法的搜索优势, 提出了一种改进型边缘采样哈里斯·鹰算法(EASS-HHO), 并以该算法与原始 HHO 算法对所建模型进行求解。实验结果表明, EASS-HHO 在求解效果和优化性能方面均优于传统 HHO 算法, 尤其是在其所包含的操作机制中, “交叉算子 2” 展现出最优的收敛性与稳定性。最后, EASS-HHO 被成功应用于实际案例中, 验证了其有效性与实用价值。

在模型构建方面, 未来的研究将进一步引入现实场景中服务时间窗的多样性, 作为服务类别差异的一种体现, 从而使问题建模更贴近真实应用场景, 提升模型的适用性和解释力。在求解算法方面, 进一步探索平衡全局搜索和局部搜索的方法, 使得算法在求解实际模型时能得到好的优化效果。

参考文献

- [1] Bae, H. and Moon, I. (2016) Multi-Depot Vehicle Routing Problem with Time Windows Considering Delivery and Installation Vehicles. *Applied Mathematical Modelling*, **40**, 6536-6549. <https://doi.org/10.1016/j.apm.2016.01.059>
- [2] Ali, O., Côté, J. and Coelho, L.C. (2021) Models and Algorithms for the Delivery and Installation Routing Problem. *European Journal of Operational Research*, **291**, 162-177. <https://doi.org/10.1016/j.ejor.2020.09.011>
- [3] 庞海军, 丁以中. 基于软时间窗的产品配送与安装相分离的车辆调度优化[J]. 上海海事大学学报, 2012, 33(1): 20-25.

- [4] Karimi, L. and Nawrin Ferdous, C. (2022) Branch and Price Algorithm for Multi-Trip Vehicle Routing with a Variable Number of Wagons and Time Windows. *Algorithms*, **15**, Article 412. <https://doi.org/10.3390/a15110412>
- [5] Hu, J., Huang, Y. and Hu, Z.A. (2021) Multi-Depot Vehicle Routing Problem with Cooperative Transportation in Terminal Distribution. *International Journal of Applied Decision Sciences*, **14**, 588-614. <https://doi.org/10.1504/ijads.2021.117473>
- [6] Hintsch, T. (2021) Large Multiple Neighborhood Search for the Soft-Clustered Vehicle-Routing Problem. *Computers & Operations Research*, **129**, Article ID: 105132. <https://doi.org/10.1016/j.cor.2020.105132>
- [7] Hernández-Pérez, H. and Salazar-González, J. (2022) A Branch-and-Cut Algorithm for the Split-Demand One-Commodity Pickup-and-Delivery Travelling Salesman Problem. *European Journal of Operational Research*, **297**, 467-483. <https://doi.org/10.1016/j.ejor.2021.05.040>
- [8] Park, H., Son, D., Koo, B. and Jeong, B. (2021) Waiting Strategy for the Vehicle Routing Problem with Simultaneous Pickup and Delivery Using Genetic Algorithm. *Expert Systems with Applications*, **165**, Article ID: 113959. <https://doi.org/10.1016/j.eswa.2020.113959>
- [9] Goel, R. and Maini, R. (2018) A Hybrid of Ant Colony and Firefly Algorithms (HAFA) for Solving Vehicle Routing Problems. *Journal of Computational Science*, **25**, 28-37. <https://doi.org/10.1016/j.jocs.2017.12.012>
- [10] Ferreira, K.M. and de Queiroz, T.A. (2018) Two Effective Simulated Annealing Algorithms for the Location-Routing Problem. *Applied Soft Computing*, **70**, 389-422. <https://doi.org/10.1016/j.asoc.2018.05.024>
- [11] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H. (2019) Harris Hawks Optimization: Algorithm and Applications. *Future Generation Computer Systems*, **97**, 849-872. <https://doi.org/10.1016/j.future.2019.02.028>
- [12] Yıldız, B.S. and Yıldız, A.R. (2019) The Harris Hawks Optimization Algorithm, Salp Swarm Algorithm, Grasshopper Optimization Algorithm and Dragonfly Algorithm for Structural Design Optimization of Vehicle Components. *Materials Testing*, **61**, 744-748. <https://doi.org/10.3139/120.111379>
- [13] Jia, H., Lang, C., Oliva, D., Song, W. and Peng, X. (2019) Dynamic Harris Hawks Optimization with Mutation Mechanism for Satellite Image Segmentation. *Remote Sensing*, **11**, Article 1421. <https://doi.org/10.3390/rs11121421>
- [14] Zhang, W., Gen, M. and Jo, J. (2013) Hybrid Sampling Strategy-Based Multiobjective Evolutionary Algorithm for Process Planning and Scheduling Problem. *Journal of Intelligent Manufacturing*, **25**, 881-897. <https://doi.org/10.1007/s10845-013-0814-2>
- [15] Marius, M.S. (2005) Solomon's Benchmark Problems. <https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/>
- [16] Rattanamane, T. and Nanthavanij, S. (2022) Heuristic Procedure for Bi-Capacitated Multiple-Trip Vehicle Routing Problem. *European Journal of Industrial Engineering*, **16**, 294-316. <https://doi.org/10.1504/ejie.2022.121904>
- [17] 李珍萍, 张煜炜. 带时间窗和服务顺序约束的多需求车辆路径问题[J]. 控制与决策, 2019, 34(7): 1565-1570.
- [18] Ishibuchi, H., Yoshida, T. and Murata, T. (2003) Balance between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling. *IEEE Transactions on Evolutionary Computation*, **7**, 204-223. <https://doi.org/10.1109/tevc.2003.810752>
- [19] 王付宇, 汤涛, 李艳, 王小牛. 疫情事件下多灾点应急资源最优化配置研究[J]. 复杂系统与复杂性科学, 2021, 18(1): 53-62.
- [20] 刘凡, 张惠珍, 周迅. 带模糊需求的开放式选址路径问题的混合离散蘑菇繁殖算法[J]. 计算机应用研究, 2021, 38(3): 738-744, 750.