

# 控制与决策

Control and Decision

## 动态扰动下云制造服务组合的区间多目标优化方法

张晓冬, 燕洁晨, 孙家正

引用本文:

张晓冬, 燕洁晨, 孙家正. 动态扰动下云制造服务组合的区间多目标优化方法[J]. *控制与决策*, 2026, 41(1): 19-30.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2025.0771>

---

### 您可能感兴趣的其他文章

#### Articles you may be interested in

##### [基于正态云模型的状态转移算法求解多目标柔性作业车间调度问题](#)

State transition algorithm based on normal cloud model for solving multiobjective flexible job shop scheduling problem

*控制与决策*. 2021, 36(5): 1181-1190 <https://doi.org/10.13195/j.kzyjc.2019.1233>

##### [基于局部搜索的反向学习竞争粒子群优化算法](#)

Opposition-based learning competitive particle swarm optimizer with local search

*控制与决策*. 2021, 36(4): 779-789 <https://doi.org/10.13195/j.kzyjc.2019.1150>

##### [基于鲁棒优化的云医疗资源配置问题](#)

Robust optimization based medical resource allocation problem in cloud healthcare system

*控制与决策*. 2021, 36(2): 469-474 <https://doi.org/10.13195/j.kzyjc.2019.0455>

##### [基于自适应正态云模型的灰狼优化算法](#)

Grey wolf optimization algorithm based on adaptive normal cloud model

*控制与决策*. 2021, 36(10): 2562-2568 <https://doi.org/10.13195/j.kzyjc.2020.0233>

##### [基于知识粒度特征的多目标粗糙集属性约简算法](#)

Multi objective rough set attribute reduction algorithm based on characteristics of knowledge granularity

*控制与决策*. 2021, 36(1): 196-205 <https://doi.org/10.13195/j.kzyjc.2019.0490>

# 动态扰动下云制造服务组合的区间多目标优化方法

张晓冬<sup>†</sup>, 燕洁晨, 孙家正

(北京科技大学 经济管理学院, 北京 100083)

**摘要:** 为解决同时存在云服务服务质量 (QoS) 属性不确定性与紧急任务扰动双重挑战的云制造服务组合优化问题, 采用区间数描述不确定的 QoS 属性信息, 建立包含两个子模型的区间多目标云服务重组优化模型. 针对模型中目标函数及约束存在区间数的特征, 提出一种融合强化学习的区间快速非支配排序遗传算法 (RINSGA-II). 在算法中, 设计基于多种启发式规则的混合初始化策略以提高初始解集质量和算法收敛速度; 为准确比较不同解的优劣, 提出结合区间数运算的区间 Pareto 支配关系和拥挤距离计算方法; 同时, 设计基于  $Q$  学习的自适应参数调整策略, 以平衡算法的全局和局部搜索能力. 最后, 基于不同规模的问题算例进行仿真实验, 结果表明, RINSGA-II 算法在所求最优解集的收敛性和多样性方面显著优于对比算法, 能够得到满足约束的鲁棒性较强的云服务组合方案.

**关键词:** 云制造; 服务组合; 动态扰动; 区间;  $Q$  学习; 多目标优化

**中图分类号:** TP391 **文献标志码:** A

**DOI:** 10.13195/j.kzyjc.2025.0771

**引用格式:** 张晓冬, 燕洁晨, 孙家正. 动态扰动下云制造服务组合的区间多目标优化方法 [J]. 控制与决策, 2026, 41(1): 19-30.

## Interval multi-objective optimization for cloud manufacturing service composition under dynamic disturbances

ZHANG Xiao-dong<sup>†</sup>, YAN Jie-chen, SUN Jia-zheng

(School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China)

**Abstract:** To solve the dual challenges of uncertain quality of service (QoS) attributes and urgent task disturbances in cloud manufacturing service composition optimization, the interval method is employed to describe the uncertainty of QoS attributes. Then, an interval multi-objective service recomposition optimization model consisting of two sub-models is established. To target the interval-number characteristics in both the objective functions and constraints of the model, a reinforcement learning-based interval fast non-dominated sorting genetic algorithm (RINSGA-II) is proposed. In the RINSGA-II, a hybrid initialization strategy based on multiple heuristic rules is designed to enhance the quality of the initial solution set and accelerate convergence. Additionally, an interval Pareto dominance and crowding distance calculation method incorporating interval arithmetic is employed to compare solution quality precisely. To balance the algorithm's global exploration and local exploitation capabilities, an adaptive parameter adjustment strategy based on  $Q$ -learning is developed. Finally, simulation experiments are conducted using problem instances of varying scales. The experimental results show that the RINSGA-II significantly outperforms comparative algorithms in terms of the convergence and diversity of the optimal solution set, and it can generate constraint-satisfying and robust service composition schemes.

**Keywords:** cloud manufacturing; service composition; dynamic disturbances; interval;  $Q$ -learning; multi-objective

## 0 引言

云制造 (CMfg) 是一种面向服务的、高效节能的网络化智能制造新模式, 能够为用户提供可随获、按需使用的各类云服务<sup>[1]</sup>. 云服务组合优化

(SCO) 是 CMfg 系统的核心功能, 旨在将细粒度的云服务组合起来共同执行用户的复杂任务, 对实现制造资源优化配置发挥着重要作用<sup>[2]</sup>.

近年来, 国内外学者对 SCO 问题的研究已取得

收稿日期: 2025-07-22; 录用日期: 2025-10-10.

基金项目: 教育部人文社会科学研究项目 (23YJA630051).

责任编辑: 巩敦卫.

<sup>†</sup>通信作者. E-mail: xdzhang@manage.ustb.edu.cn.

了一系列成果. 例如, Wang 等<sup>[3]</sup>综合考虑了云平台、资源提供商和用户的利益, 建立了八目标的云服务组合优化模型; Wan 等<sup>[4]</sup>针对多重组合任务的制造服务分配问题, 提出了一种分层调度模型; Tong 等<sup>[5]</sup>研究了产品交付及时性对用户满意度的影响, 构建了以满意度、能耗和成本为目标的调度模型并设计了改进的混合人工蜂群算法. 上述研究对于解决 SCO 问题具有重要指导意义, 但是, 在问题模型的构建过程中都将云服务的服务质量 (QoS) 属性值按照标准值进行处理, 并基于此求解模型寻求最优的云服务组合方案.

实际的 CMfg 环境存在多种不确定因素, 诸如员工及设备造成的误差、用户需求变动、市场不稳定等, 这些不确定因素不可避免地会导致云服务的服务时间、服务成本等 QoS 属性呈现出区间波动性<sup>[6-7]</sup>. 在该情况下, 若忽略这些不确定因素, 则将导致基于标准值生成的云服务组合在实际执行中的性能指标发生恶化. 因此, 有必要在 SCO 问题中预先考虑 QoS 属性不确定性的影响, 进而提高服务组合方案的鲁棒性及 CMfg 系统的稳定性.

此外, 服务组合方案在执行过程中还会受到其他不可预知的动态事件干扰, 如紧急任务扰动、制造服务故障等<sup>[8]</sup>, 在动态事件发生时对服务组合进行重构是处理此类意外干扰的有效方法. 例如, Elgendy 等<sup>[9]</sup>针对新任务到达的情况, 提出了一个混合整数数学模型并设计了并行分布式遗传算法; 王彦凯等<sup>[10]</sup>研究了具有多种实际约束以及服务异常情况的服务组合问题, 提出了一种服务组合的自适应重构模型; Dai 等<sup>[11]</sup>提出了考虑多供应链协作的新任务到达的协同任务调度模型, 并设计了改进的多种群生物地理优化算法; 董明等<sup>[12]</sup>针对订单动态到达平台的情况, 在静态策略基础上建立了以延期时间最小为目标的调度模型. 上述研究对于 SCO 问题中的动态扰动都提出了有效的解决方案, 但仍存在以下不足: 在生成初始服务组合方案以及建立服务组合重构模型时, 仍是基于确定的云服务 QoS 属性值, 没有考虑到 CMfg 中固有的 QoS 属性的波动性. 因而, 在该情况下求解模型得到的重组方案在实际执行中的性能指标仍可能恶化.

在 SCO 问题中, 有必要同时考虑云服务 QoS 属性的不确定性和其他动态干扰对服务组合的影响. 目前对于此问题可参考的研究较少, Zhang 等<sup>[13]</sup>同时考虑了原材料交付时间的不确定性和紧急任务扰动, 建立了两阶段服务组合模型. 然而, 该研究将模型简化为单目标问题进行求解, 不适用于实际生产

中更为复杂的多目标优化需求.

SCO 问题是一类具有 NP-hard 复杂性的多目标优化问题, 元启发式算法仍是目前求解此类问题的主流技术<sup>[14]</sup>. 近年来, 针对多目标优化模型, 研究者开始采用将强化学习与元启发式算法相结合的方法, 以增强算法的搜索效率与适应性<sup>[15-16]</sup>. 与传统元启发式算法相比, 融合强化学习能够利用环境反馈引导算法搜索方向, 从而有效提升算法求解性能. 因此, 本文采用结合强化学习的多目标进化算法来求解模型, 并结合实际问题特征对算法进行针对性改进, 使其能够更高效地求解动态多目标 SCO 问题.

基于上述研究存在的不足, 本文对同时存在 QoS 属性不确定性和紧急任务干扰的多任务 SCO 问题进行研究: 首先, 采用区间数处理 QoS 属性的不确定性, 建立区间多目标云服务重组优化模型; 然后, 针对模型中目标函数及约束存在区间数的特征, 提出一种融合强化学习的区间快速非支配排序遗传算法; 最后, 基于不同尺度的实验算例, 对模型和算法的有效性进行验证.

## 1 问题描述与建模

### 1.1 问题描述

云服务组合过程涉及资源提供方、云服务需求方和云平台 3 个角色, 具体如图 1 所示. 从需求方提交制造任务请求到任务完成, 需要经历以下阶段:

1) 任务分解. 对于云平台在某一决策周期内接收到的任务, 可表示为  $\text{Task} = \{T_1, T_2, \dots, T_i, \dots, T_n\}$ . 假定每一任务由不同的需求方提供, 需要满足不同的时间、成本等 QoS 约束. 对于任务  $T_i$ , 云平台将按照该任务的工艺流程将其分解为一系列单一资源需求的子任务, 即  $T_i = \{ST_{i1}, ST_{i2}, \dots, ST_{ij}, \dots, ST_{ik_i}\}$ . 同一任务的各个子任务之间存在执行顺序, 不同任务的子任务之间彼此独立.

2) 云服务搜索匹配. 云平台从资源池中为每个  $ST_{ij}$  匹配满足其功能需求的可用云服务, 形成  $ST_{ij}$  的候选云服务集  $\text{CSS}_{ij} = \{\text{CS}_{ij}^1, \text{CS}_{ij}^2, \dots, \text{CS}_{ij}^l, \dots, \text{CS}_{ij}^{L_{ij}}\}$ , 其中  $\text{CS}_{ij}^l$  为  $ST_{ij}$  的第  $l$  个候选云服务. 同一  $\text{CSS}_{ij}$  内的各个服务功能相同但 QoS 不同, 每个云服务均由实体的资源提供商提供, 每个提供商可提供多种不同功能的云服务.

3) 服务组合优化. 对于任意  $T_i$ , 云平台分别从其各个子任务的候选云服务集中选择一个云服务进行组合, 生成  $T_i$  所有可能的服务组合方案. 多任务 SCO 旨在根据不同用户的个性化需求, 为每个任务选择合适的云服务组合, 使得所有任务的服务组合

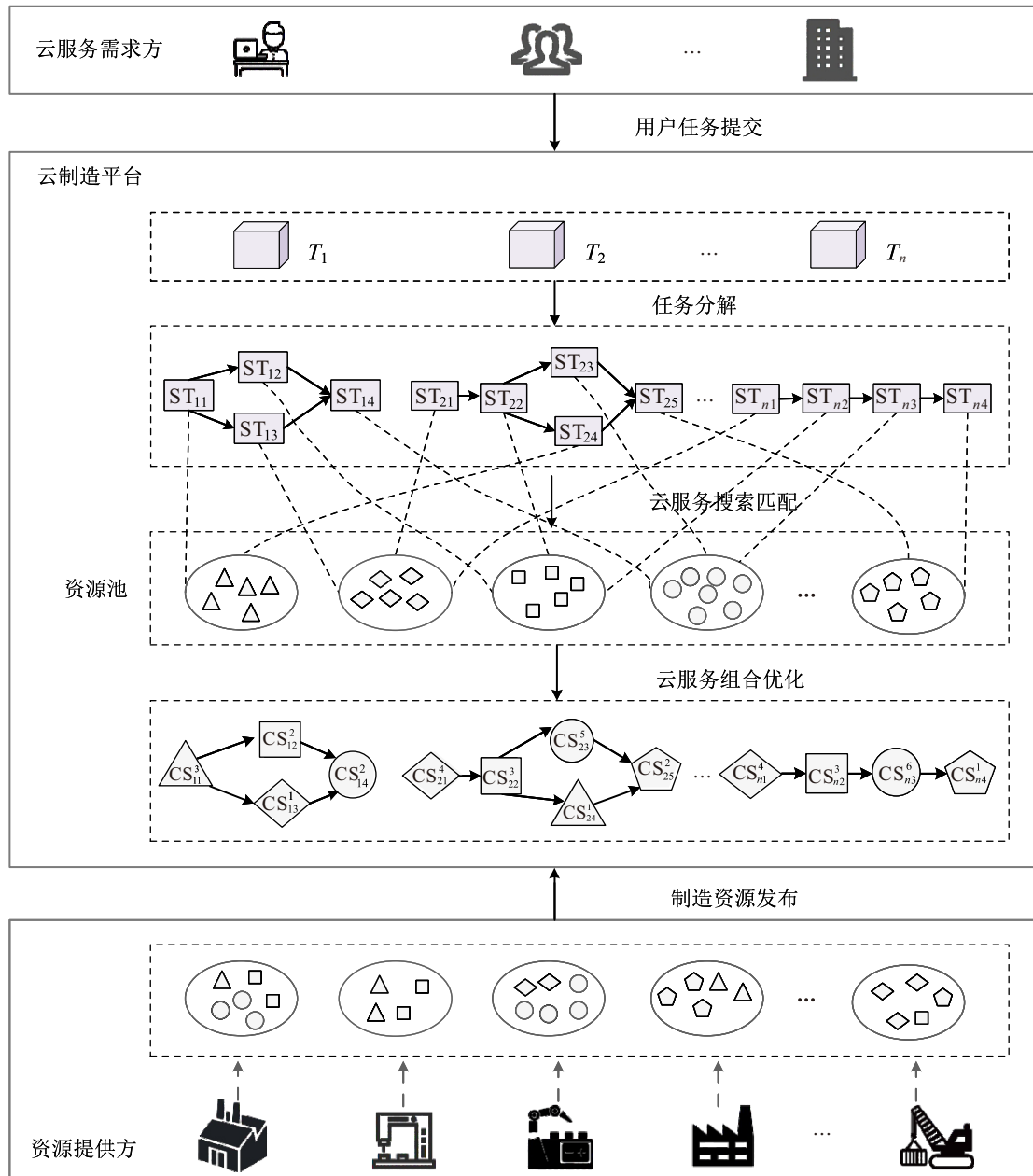


图1 云制造服务组合过程

的整体 QoS 指标最优.

4) 执行与反馈. 云平台对服务组合执行全过程进行监管, 并对可能出现的异常进行检测与修复.

本文首先考虑云服务的服务时间、服务成本属性的不确定性, 并通过区间数表示不确定参数的取值. 基于云平台的历史数据记录, 可以统计云服务加工同类型子任务的服务时间和服务成本的最大值和最小值, 进而得到区间的上下界. 其次, 考虑服务组合执行过程中的某一时刻, 云平台接收到紧急任务的情况. 当出现紧急任务扰动时, 云平台需要及时对初始服务组合进行调整以应对扰动. 此外, 当同一任务的相邻子任务所选择的云服务由不同的提供商提供时, 还需要考虑物流时间及物流成本.

本文的研究目标为: 在多种动态扰动并存的

CMfg 环境下合理调控云服务资源, 为各个子任务分配合适的云服务, 使得所有任务的云服务组合方案集合的整体性能指标达到最优. 为便于建立模型, 作出如下假设:

- 1) 不同用户任务之间相互独立, 除紧急任务外, 其他任务的优先级相同;
- 2) 云平台已完成任务分解和云服务搜索匹配;
- 3) 每个云服务可以完成多种类型的子任务, 但在同一时刻只能处理一项子任务;
- 4) 每个子任务只能由候选云服务集合内的一个云服务处理;
- 5) 在初始时刻, 所有服务均可用;
- 6) 所有子任务一旦在所选云服务上开始处理, 则不能中断.

## 1.2 符号定义

为便于描述模型, 定义符号如表 1 所示.

表1 模型相关符号定义

符号	定义
$i$	任务编号, $i \in I = \{1, 2, \dots, n\}$
$j$	子任务编号, $j \in J = \{1, 2, \dots, J_i\}$
$R$	紧急任务, $R = \{1, 2, \dots, r\}$
$I'$	未完成的任务, $I' \subseteq I$
$J'$	任务 $i$ 未完成的子任务, $J' \subseteq J$
$k$	云服务编号, $k \in K = \{1, 2, \dots, s\}$ , $MS_k$ 表示第 $k$ 个云服务
$m$	服务提供商编号, $m \in M = \{1, 2, \dots, p\}$ , $SP_m$ 表示第 $m$ 个服务提供商
$\tilde{t}_{ij}^k$	子任务 $ST_{ij}$ 在 $MS_k$ 上所需的服务时间, $\tilde{t}_{ij}^k = [t_{ij}^{k-}, t_{ij}^{k+}]$
$\tilde{c}_{ij}^k$	子任务 $ST_{ij}$ 在 $MS_k$ 上所需的服务成本, $\tilde{c}_{ij}^k = [c_{ij}^{k-}, c_{ij}^{k+}]$
$tr_{mm'}$	从 $SP_m$ 到 $SP_{m'}$ 所需的物流时间, 若 $m = m'$ , 则 $tr_{mm'} = 0$
$cr_{mm'}$	从 $SP_m$ 到 $SP_{m'}$ 所需的物流成本, 若 $m = m'$ , 则 $cr_{mm'} = 0$
$Z_{ij}^k$	若 $ST_{ij}$ 能够由 $MS_k$ 完成, 则 $Z_{ij}^k = 1$ ; 否则 $Z_{ij}^k = 0$
$R_{ijk}^m$	若 $ST_{ij}$ 所选择的第 $k$ 个云服务 $MS_k$ 由 $SP_m$ 提供, 则 $R_{ijk}^m = 1$ ; 否则 $R_{ijk}^m = 0$
$\tilde{S}_{ij}^k$	$ST_{ij}$ 在 $MS_k$ 上的开始时间
$\tilde{F}_{ij}^k$	$ST_{ij}$ 在 $MS_k$ 上的完成时间
$\tilde{T}_i$	第 $i$ 个任务的完成时间
$\tilde{C}_i$	第 $i$ 个任务所需的总成本
$T_{\max}^i$	第 $i$ 个任务的允许完成时间上限
$C_{\max}^i$	第 $i$ 个任务的总成本上限
$\delta$	拖期的单位惩罚成本
$P$	足够大的正数
$\tilde{T}$	所有任务的完成时间
$\tilde{C}$	所有任务的总成本
$x_{ij}^k$	若 $ST_{ij}$ 被分配到第 $k$ 个云服务上, 则 $x_{ij}^k = 1$ ; 否则 $x_{ij}^k = 0$
$y_{i_1 j_1, i_2 j_2}^k$	若 $ST_{i_1 j_1}$ 和 $ST_{i_2 j_2}$ 都在 $MS_k$ 上, 且 $ST_{i_1 j_1}$ 在 $ST_{i_2 j_2}$ 之前被处理, 则 $y_{i_1 j_1, i_2 j_2}^k = 1$ ; 否则 $y_{i_1 j_1, i_2 j_2}^k = 0$

## 1.3 区间多目标云服务重组优化模型

### 1.3.1 区间云服务组合优化模型

首先考虑云服务的服务时间、服务成本的不确定性, 建立区间云服务组合优化模型 M1, 以生成各任务的初始服务组合方案. 目标函数如下:

1) 最小化总完成时间: 总完成时间  $\tilde{T}$  取决于所有任务中最后一个完工的子任务的完成时间  $\tilde{F}_{iJ_i}^k$ , 即

$$\min \tilde{T} = \max_{1 \leq i \leq n} \tilde{T}_i, \quad \tilde{T}_i = \tilde{F}_{iJ_i}^k. \quad (1)$$

2) 最小化总成本: 总成本  $\tilde{C}$  包括所有任务服务

组合的总服务成本  $\tilde{C}_p$  和总运输成本  $C_t$ , 即

$$\min \tilde{C} = \tilde{C}_p + C_t; \quad (2)$$

$$\tilde{C}_p = \sum_{i=1}^n \sum_{j=1}^{J_i} \sum_{k=1}^s x_{ij}^k \tilde{c}_{ij}^k,$$

$$C_t = \sum_{i=1}^n \sum_{j=1}^{J_i-1} \sum_{k=1}^s \sum_{m=1}^p \sum_{k'=1}^s \sum_{m'=1}^p x_{ij}^k R_{ijk}^m x_{i,j+1}^{k'} R_{i,j+1,k'}^{m'} cr_{mm'}.$$

模型的约束条件包括

$$\sum_{k=1}^s x_{ij}^k = 1, \quad i \in I, j \in J. \quad (3)$$

$$\sum_{k=1}^s x_{ij}^k Z_{ij}^k = 1, \quad i \in I, j \in J. \quad (4)$$

$$\tilde{F}_{ij}^k = \tilde{S}_{ij}^k + \tilde{t}_{ij}^k, \quad i \in I, j \in J. \quad (5)$$

$$\tilde{S}_{ij}^k \geq \begin{cases} 0, & j = 1; \\ \tilde{F}_{i,j-1}^{k'} + \sum_{k'=1}^s \sum_{m=1}^p \sum_{k=1}^s \sum_{m'=1}^p x_{i,j-1}^{k'} R_{i,j-1,k'}^m x_{ij}^k R_{ijk}^{m'} tr_{mm'}, & j > 1. \end{cases} \quad (6)$$

$$x_{i_1 j_1}^k + x_{i_2 j_2}^k - 1 \leq y_{i_1 j_1, i_2 j_2}^k + y_{i_2 j_2, i_1 j_1}^k \leq 1, \quad i_1, i_2 \in I, j_1, j_2 \in J, k \in K. \quad (7)$$

$$\tilde{S}_{i_2 j_2}^k \geq \tilde{F}_{i_1 j_1}^k - P(3 - y_{i_1 j_1, i_2 j_2}^k - x_{i_1 j_1}^k - x_{i_2 j_2}^k), \quad i_1, i_2 \in I, j_1, j_2 \in J, k \in K. \quad (8)$$

$$\tilde{T}_i \leq T_{\max}^i, \quad i \in I. \quad (9)$$

$$\tilde{C}_i \leq C_{\max}^i, \quad i \in I. \quad (10)$$

约束 (3) 表示每个子任务只能选择一个云服务; 约束 (4) 保证了被选择的云服务在该子任务的候选云服务集中; 约束 (5) 为完成时间约束; 约束 (6) 表示同一任务的任意两个相邻子任务, 后者的开始时间取决于前一子任务的完成时间和两个子任务间的运输时间; 约束 (7) 和 (8) 表示在同一云服务上的两个子任务, 只有当前的子任务完成后才能开始处理下一个; 约束 (9) 和 (10) 表示任务的完成时间和总成本不能超过用户提出的最大限制.

### 1.3.2 云服务重组优化模型

出现紧急任务扰动时, 云平台需要对未完成的任务及紧急任务进行服务重组. 重组过程仍需考虑云服务 QoS 属性的不确定性对模型的影响. 建立云服务重组优化模型 M2, 目标函数如下:

1) 最小化总完成时间

$$\min \tilde{T}, \tilde{T} = \max_{i \in I' \cup R} \tilde{T}_i. \quad (11)$$

2) 最小化总成本: 云平台为保证紧急任务按时交付, 会优先将云服务分配给紧急任务, 进而导致初始的部分任务可能延期完成. 当任务完成时间超过需求方规定的交付时间时, 平台需要支付给用户相应的惩罚成本. 为保证紧急任务按时完成, 将其惩罚系数设置为充分大的正数. 因此, 总成本包括  $\tilde{C}_p$ 、 $C_t$  及惩罚成本  $\tilde{C}_d$  三部分.

$$\min \tilde{C} = \tilde{C}_p + C_t + \tilde{C}_d. \quad (12)$$

$$\tilde{C}_p = \sum_{i \in I' \cup R} \sum_{j \in J'} \sum_{k \in K} x_{ij}^k \tilde{c}_{ij}^k;$$

$$C_t = \sum_{i \in I' \cup R} \sum_{j \in J'} \sum_{k \in K} \sum_{m \in M} \sum_{k' \in K} \sum_{m' \in M} x_{ij}^k R_{ijk}^m x_{i,j+1}^{k'} R_{i,j+1,k'}^{m'} cr_{mm'};$$

$$\tilde{C}_d = \sum_{i \in I' \cup R} \tilde{d}_i \delta, \quad \tilde{d}_i = \begin{cases} \tilde{T}_i - T_{\max}^i, & \tilde{T}_i \geq T_{\max}^i; \\ 0, & \text{otherwise.} \end{cases}$$

3) 最小化方案偏差值: 方案偏差值可以用未完成的子任务的云服务变动的累积来计算. 当子任务选择的云服务发生变更时也会影响该子任务与其相邻子任务间的运输成本及运输时间.  $x_{ij}^k$  为  $ST_{ij}$  的初始服务选择,  $x_{ij}^{k'}$  为  $ST_{ij}$  在重组阶段的服务选择.

$$\min DE = \frac{1}{2} \sum_{i \in I'} \sum_{j \in J'} \sum_{k \in K} |x_{ij}^k - x_{ij}^{k'}|. \quad (13)$$

M1 中的约束条件均同样适用于 M2, 限于篇幅, 不再重述.

### 1.4 区间数运算操作

对区间数的相关运算描述如下:

1) 加法运算. 用于计算模型的目标函数值. 对于任意两个区间数  $\tilde{a} = [a^-, a^+]$  和  $\tilde{b} = [b^-, b^+]$ ,  $\tilde{a} + \tilde{b} = [a^- + b^-, a^+ + b^+]$ .

2) 区间取大. 用于计算子任务在所选云服务上的最早可用时间,  $\tilde{a} \vee \tilde{b} = [\max(a^-, b^-), \max(a^+, b^+)]$ .

3) 区间数排序. 为提高区间数排序的准确性和合理性, 以便更好地比较不同解的目标值, 引入三角分布下区间数排序的可能度计算模型<sup>[17]</sup>. 以目标  $\tilde{T}$  为例, 构建三角分布  $\tilde{T} \sim \Delta(T^-, \tau, T^+)$ , 其中众数  $\tau$  表示区间内最可能的取值. 区间数比较准则如下 (其中  $P$  为可能度,  $P\{\tilde{a} \geq \tilde{b}\}$  表示区间  $\tilde{a}$  大于区间  $\tilde{b}$  的程度):

①  $P\{\tilde{a} \geq \tilde{b}\} = 0.5$ ,  $\tilde{a}$  与  $\tilde{b}$  拟相等, 记作  $\tilde{a} \approx \tilde{b}$ ;

②  $P\{\tilde{a} \geq \tilde{b}\} > 0.5$ ,  $\tilde{a}$  拟大于  $\tilde{b}$ , 记作  $\tilde{a} \succ \tilde{b}$ .

## 2 算法设计

针对模型多目标且目标函数及约束存在区间数

的特征, 本文提出一种基于强化学习的区间快速非支配排序遗传算法 (RINSGA-II). 算法结合问题特征设计了混合初始化策略生成初始种群, 以兼顾初始解的质量和多样性; 采用融合区间数运算的 Pareto 支配和拥挤距离; 提出基于  $Q$ -learning 的自适应参数调整策略, 以平衡算法的全局与局部搜索能力.

### 2.1 染色体编码

采用双层整数编码方式对所有任务的云服务组合进行编码, 如图 2 所示. 染色体包括两部分:

1) 第 1 层表示子任务间的处理顺序信息. 每个基因位点的值用任务索引表示, 从左到右, 用任务索引出现的次数表示该任务的子任务索引. 例如, 该片段中第 1 个“1”表示任务  $T_1$  的第 1 个子任务, 第 2 个“1”表示  $T_1$  的第 2 个子任务.

2) 第 2 层表示子任务的云服务选择信息. 每个基因位点的值表示子任务从其候选云服务集中选择的云服务编号. 染色体第 2 层与第 1 层一一对应, 如  $ST_{22}$  和  $ST_{33}$  都在  $MS_6$  上处理且  $ST_{22}$  优于  $ST_{33}$ .

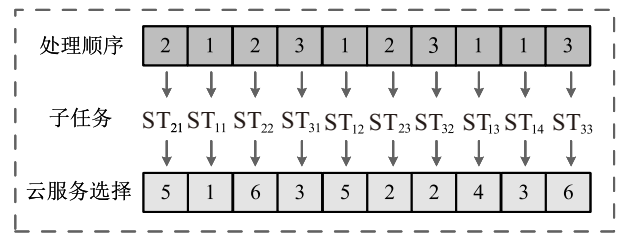


图2 染色体编码

### 2.2 基于启发式规则的混合初始化策略

首先, 对于染色体的云服务选择部分, 采取以下 4 个初始化规则:

1) 最小完成时间规则: 对于子任务  $ST_{ij}$ , 从其候选云服务集中选择  $ST_{ij}$  完成时间最早的云服务.

2) 最小服务时间规则: 对于  $ST_{ij}$ , 选择具有最短服务时间的云服务.

3) 最小服务成本规则: 对于  $ST_{ij}$ , 选择服务成本最小的云服务.

4) 随机选择: 从各子任务的候选云服务集合中随机选择云服务, 以保证解的多样性.

在确定每个子任务的云服务选择信息后, 进一步生成染色体子任务排序部分, 规则如下:

1) 最大剩余工作量规则: 对于目前所有任务, 将所需剩余服务时间最大的任务编号放入染色体编码中, 同时更新该任务的剩余工作量信息.

2) 随机规则: 将所有的子任务随机排列, 以生成该部分染色体编码.

染色体服务选择部分 4 个初始化规则被选择的

概率分别设置为 0.3、0.1、0.1、0.5; 子任务排序部分各规则被选择的概率分别为 0.4、0.6.

### 2.3 区间 Pareto 支配及拥挤距离

传统的 Pareto 支配无法直接用于比较目标值为区间数的不同解的优劣. 因此, 本文结合非均匀分布区间数排序的可能度计算模型, 提出区间 Pareto 支配关系: 对于任意两个解  $x$ 、 $y$ , 相应的目标函数值表示为  $\tilde{f}_i(x)$ 、 $\tilde{f}_i(y)$ ,  $i=1, 2, \dots, k$ , 计算  $P(\tilde{f}_i(x) \leq \tilde{f}_i(y))$ . 若对于  $\forall i, P(\tilde{f}_i(x) \leq \tilde{f}_i(y)) \geq 0.5$ , 且  $\exists i$  使得  $P(\tilde{f}_i(x) \leq \tilde{f}_i(y)) > 0.5$ , 则称  $x$  支配  $y$ .

基于确定目标值的拥挤距离同样也不再适用于本文问题, 参考文献 [18], 可采用基于区间数的拥挤距离计算方法. 对于个体  $x$ , 其拥挤距离可由与其最相邻的两个个体  $y$  和  $z$  确定, 具体如下:

step 1: 对所有个体的区间目标值归一化处理, 以区间目标  $\tilde{f}_i(x) = [f_i(x)^-, f_i(x)^+]$  为例, 归一化后可表示为  $f_i^*(x) = [f_i^*(x)^-, f_i^*(x)^+]$ .

$$f_i^*(x)^- = \frac{f_i(x)^- - f_i^{\min}(X)^-}{f_i^{\max}(X)^+ - f_i^{\min}(X)^-}, \quad (14)$$

$$f_i^*(x)^+ = \frac{f_i(x)^+ - f_i^{\min}(X)^-}{f_i^{\max}(X)^+ - f_i^{\min}(X)^-}. \quad (15)$$

step 2: 式 (16) 表示个体  $x$  的拥挤距离, 其中  $x$  与  $y$  之间的距离

$$D(x) = \frac{d(x, y) + d(x, z)}{2}; \quad (16)$$

$$d(x, y) = \frac{\sum_{i=1}^k |\text{mid}(f_i^*(x)) - \text{mid}(f_i^*(y))|}{\gamma(x, y) + V(x) + V(y) + 1}, \quad (17)$$

$$\gamma(x, y) = \prod_{i=1}^k w(f_i^*(x) \cap f_i^*(y)), \quad (18)$$

$$V(x) = \prod_{i=1}^k w(f_i^*(x)). \quad (19)$$

其中:  $\text{mid}(f_i^*(x))$  表示  $x$  的第  $i$  个目标分量值归一化后的区间中点;  $\gamma(x, y)$  表示  $x$  与  $y$  对应超立方体的重叠度,  $V(x)$  为  $x$  对应的超立方体的体积, 可以用各区间目标的宽度的乘积表示,  $\gamma(x, y)$  和  $V(x)$  仅适用于区间目标值;  $f_i^*(x) \cap f_i^*(y)$  表示两区间数的交区间,  $w(f_i^*(x) \cap f_i^*(y))$  表示其宽度. 此外, 令边界解的拥挤距离为无穷大, 以增大其进入下一代的概率.

## 2.4 遗传操作

### 2.4.1 选择算子

为兼顾算法的收敛速度和解的多样性, 采用结合随机选择和二元锦标赛的方法, 步骤如下:

step 1: 随机生成  $p \in (0, 1)$ , 若  $p < p_r$  ( $p_r$  为随机选择的概率,  $p_r = 0.6$ ), 则随机选择一个个体; 否则采用二元锦标赛法, 转 step 2.

step 2: 从种群中随机选择两个个体, 比较其非支配排序号, 优先选择排序号较低的个体; 若处于同一等级, 则优先选择拥挤距离较大的个体.

step 3: 重复上述步骤, 直至选出  $N$  个个体.

### 2.4.2 交叉算子

1) 染色体云服务选择部分采用均匀交叉: 对于父代染色体  $P_1$  和  $P_2$ , 随机生成  $\sum_{i=1}^n J_i$  个在  $(0, 1)$  上服从均匀分布的随机数; 将  $P_1$  中  $\rho > 0.5$  的位置对应的子任务的云服务与  $P_2$  中相同子任务的云服务交换; 染色体子任务排序部分保持不变.

2) 染色体子任务排序部分采用 IPOX 交叉: 将任务集 Task 随机划分成两个非空子集  $T_1$  和  $T_2$ , 且  $T_1 \cap T_2 = \emptyset$ ; 将  $P_1$  中属于  $T_1$  的基因复制到子代染色体  $C_1$  中, 将  $P_2$  中属于  $T_2$  的基因复制到  $C_2$  中, 保留上述基因的位置; 然后, 再分别将  $P_1$  中属于  $T_1$  的基因和  $P_2$  中属于  $T_2$  的基因按顺序依次填充到  $C_2$  和  $C_1$  中; 各子任务选择的云服务保持不变.

### 2.4.3 变异算子

染色体子任务排序部分采用两点变异: 对于父代染色体, 随机选择两个变异位点  $m_1$  和  $m_2$ ,  $m_1$  和  $m_2$  处的基因值不能相同, 然后将  $m_1$  和  $m_2$  处的子任务互换.

染色体服务选择部分采用多点变异: 随机选择  $m$  个变异位点 ( $0 < m \leq n$ ,  $n$  为任务数), 对于每个变异位点, 从其对应的子任务的候选云服务集中随机选取一个新的云服务替换当前云服务.

## 2.5 基于 Q-learning 的自适应参数调整

Q-learning 是一种经典的无模型强化学习算法, 旨在通过 Agent 与环境交互来迭代更新  $Q$  表, 以学习最优策略. Q-learning 中, Agent 在  $t$  时刻可感知当前所处的状态  $s_t$ , 根据策略选择动作  $a_t$ , 执行  $a_t$  后得到环境的即时反馈奖励  $r_{t+1}$ , 并转移到新的状态  $s_{t+1}$ , 进而更新  $Q$  值, 有  $Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t))$ . 其中:  $\alpha$  为学习率,  $\gamma$  为折扣因子.

为平衡算法的全局和局部搜索能力, 在 RINSGA-II 中嵌入 Q-learning 算法. 通过更新  $Q$  表来引导算法在迭代中根据当前种群的特征自适应地选择交叉及变异概率. Q-learning 各部分设计如下:

1) 状态. 将每一代的种群看作 Agent, 选择评估解集质量的综合评价指标 HV 和多样性评价指标

Sp来描述 Agent 在环境中所处的状态, 并根据两指标的变化来划分状态空间, 如表 2 所示. HV值越大, 解集的收敛性和多样性越好; Sp值越小, 解集的分布性和多样性则越好. 计算公式如下:

$$HV(P, r) = \bigcup_{x \in P} v(x, r), \quad (20)$$

$$Sp = \sqrt{\frac{1}{|P| - 1} \sum_{i=1}^{|P|} (d_i - \bar{d})^2}. \quad (21)$$

其中:  $P$ 为当前种群中的非支配解集;  $r$ 为参考点;  $v(x, r)$ 为解 $x$ 与 $r$ 之间所形成的空间的体积;  $d_i$ 为第 $i$ 个解到 $P$ 中最邻近解的距离;  $\bar{d}$ 为所有 $d_i$ 的平均值. 为消除不同目标间量纲差异的影响, 需将目标函数值归一化处理; 同时为便于计算, 用区间的众数值来代替区间数进行计算.

表2 状态空间

状态	描述
$s_1$	$Sp_{t+1} - Sp_t < 0$ 且 $HV_{t+1} - HV_t > 0$
$s_2$	$Sp_{t+1} - Sp_t < 0$ 且 $HV_{t+1} - HV_t \leq 0$
$s_3$	$Sp_{t+1} - Sp_t \geq 0$ 且 $HV_{t+1} - HV_t > 0$
$s_4$	$Sp_{t+1} - Sp_t \geq 0$ 且 $HV_{t+1} - HV_t \leq 0$

2) 动作. 将动作定义为交叉和变异概率的调整方向, 包括增加、减少和保持不变.  $P_c$ 和 $P_m$ 计算如下:

$$P_c^{t+1} = P_c^t + \Delta P_c, \Delta P_c \in \{-0.05, 0, 0.05\}, \quad (22)$$

$$P_m^{t+1} = P_m^t + \Delta P_m, \Delta P_m \in \{-0.02, 0, 0.02\}, \quad (23)$$

其中 $P_c^t$ 和 $P_m^t$ 为第 $t$ 次迭代中 $P_c$ 和 $P_m$ 的值. 结合 $\Delta P_c$ 和 $\Delta P_m$ 的变化组合, 共有 9 个可选动作. 结合已有研究中参数取值的实践经验<sup>[9,19]</sup>, 本文分别控制 $P_c$ 和 $P_m$ 的变动范围为 $[0.65, 0.95]$ 、 $[0.01, 0.2]$ .

3) 奖励函数. 根据 Agent 执行动作后 Sp和 HV 指标值的变化来定义奖励函数. 本节采用加权线性奖励

$$r_{t+1} = w_1 \frac{Sp_t - Sp_{t+1}}{Sp_t} + w_2 \frac{HV_{t+1} - HV_t}{HV_t}. \quad (24)$$

其中:  $w_1$ 和 $w_2$ 表示指标权重,  $w_1 = 0.4$ ,  $w_2 = 0.6$ .

4) 策略. 本节设计改进的  $\epsilon$ -greedy 策略来指导 Agent 在不同状态下的动作选择. 随机生成  $\text{rand} \in (0, 1)$ , 若  $\text{rand} < \epsilon$ , 则随机选取动作; 否则选取  $Q$  值最大的动作. 使用余弦函数调节  $\epsilon$  的衰减以平衡算法的探索和利用, 即

$$\epsilon = \max \left( \epsilon_0 \cos \left( \frac{t-1}{2(\max \text{Gen} - 1)} \pi \right), 0.05 \right), \quad (25)$$

其中 $\epsilon_0$ 表示初始探索率. 在迭代初期,  $Q$ -learning 具

有较高的探索率进而能更多地探索不同的动作; 随着 $\epsilon$ 降低, Agent 在后期会更多地利用经验知识来选取  $Q$  值最大的动作, 以加快算法收敛到最优策略.

## 2.6 算法流程

本文提出的 RINSGA-II 算法流程如图 3 所示. RINSGA-II 算法对问题的求解过程包括两个阶段: 一是对模型 M1 求解, 得到所有初始任务的服务组合. 二是当有紧急任务到达云平台时, 会统计当前初始任务的执行进度, 将未开始加工的子任务及紧急任务合并, 构成新的任务集; 然后, 根据任务执行情况计算所有云服务的当前最早可用时间; 接下来, 将更新后的任务集与服务的可用时间信息更新到模型 M2 中并作为算法的输入参数; 最后, 用算法求解模型 M2, 输出所有任务的新的服务组合方案.

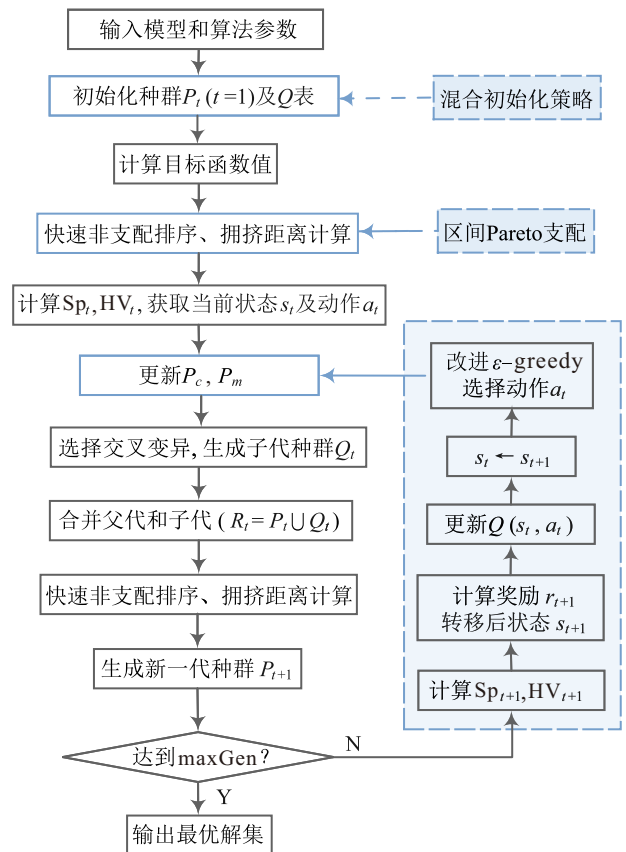


图3 RINSGA-II 算法流程

## 3 数值实验与分析

### 3.1 问题算例生成

由于云制造中对不同类型的制造服务及制造任务的描述和定义具有多样性, 目前尚未有公开的标准测试集<sup>[20]</sup>. 因此, 本文参考相关研究<sup>[8,20]</sup>, 随机生成不同尺度的服务组合测试算例展开实验, 具体如表 3 所示.

表 3 共包括 8 组算例, 每组算例分为 3 种情形, 其中 0 表示无紧急任务扰动. 具体数据包括紧急任

表3 测试算例设置

算例	$r$	At	$n$	$J_i$	SP	SP_s	$S$
1	0	0					
	3	10	5	4	6	3	6
	5	15					
2	0	0					
	3	16	10	4	8	4	6
	5	18					
3	0	0					
	3	16	15	6	13	5	9
	5	20					
4	0	0					
	3	22	20	6	15	6	9
	5	25					
5	0	0					
	3	24	25	8	20	7	12
	5	28					
6	0	0					
	3	26	30	8	23	8	12
	5	30					
7	0	0					
	3	28	35	10	30	9	15
	5	35					
8	0	0					
	3	32	40	10	32	10	15
	5	40					

务数 ( $r$ ) 及到达时间 (At)、初始任务数 ( $n$ )、每个任务的子任务数 ( $J_i$ )、云服务提供商的数量 (SP)、每个提供商可提供的云服务类型 (SP\_s)、资源池中的云服务种类 ( $S$ )。以算例 1\_3 为例, “1\_3” 表示算例 1 中有 3 个紧急任务的情形。假定各任务的子任务之间按照顺序结构执行。结合实际生产实践, 云服务相关参数的取值如表 4 所示。其中,  $\tilde{t}_{ij}^k$ 、 $\tilde{c}_{ij}^k$  对应的区间数可在  $t_{ij}^k$ 、 $c_{ij}^k$  取值的基础上, 在 5% ~ 20% 的波动范围内生成。所有资源提供商的位置坐标随机生成, 由位置坐标计算可得到  $tr_{mm'}$  和  $cr_{mm'}$  的取值。

表4 参数取值范围

参数	取值
$t_{ij}^k/h$	$U(10, 40)$
$c_{ij}^k/元$	$U(2000, 4000)$
$T_{max}^i/h$	$J_i U(35, 40)$
$C_{max}^i/元$	$J_i U(3000, 4000)$
$\delta/(元/h)$	20

### 3.2 对比算法与参数设置

为检验 RINSGA-II 算法在解决多目标动态 SCO 问题上的有效性, 选取如下算法进行对比。

1) MOEA<sup>[21]</sup>: 针对模型多目标且存在区间数特点设计的多目标进化算法, 用于验证 RINSGA-II 求解区间多目标优化模型的有效性。

2) ENSGA-II<sup>[22]</sup>: 将增强 NSGA-II 算法用于求解存在不确定加工时间和新作业插入的柔性作业车间重调度问题, 可验证 RINSGA-II 在求解具有此类扰动的 SCO 问题上的有效性。

3) IHHA<sup>[19]</sup>: 针对多目标多任务调度模型提出的改进超启发式算法, 可用于验证 RINSGA-II 对多目标 SCO 模型的求解性能。

4) EMOAHA<sup>[23]</sup>: 用于求解自适应鲁棒 SCO 模型的增强多目标人工蜂鸟算法, 可用于检验 RINSGA-II 的求解性能。

为公平比较, 将 RINSGA-II 中基于区间数的相关操作加入 ENSGA-II、IHHA 和 EMOAHA 中。算法代码均采用 MATLAB 编程实现, 所有算法种群规模均为 100, 最大迭代次数为 200。算法性能采用世代距离 (GD)、反向世代距离 (IGD)<sup>[5]</sup> 评价指标来衡量。对比算法均采用原文献中的参数。

对于 RINSGA-II 算法: 初始  $P_c$  和  $P_m$  取  $P_c = 0.8$ 、 $P_m = 0.1$ ; 对于  $Q$ -learning 中的  $\alpha$ 、 $\gamma$ 、 $\varepsilon_0$ , 采用田口法 (DOE) 设计正交实验来确定取值。各参数取值设置为  $\alpha \in \{0.3, 0.5, 0.7, 0.9\}$ ,  $\gamma \in \{0.5, 0.7, 0.8, 0.9\}$ ,  $\varepsilon_0 \in \{0.6, 0.7, 0.8, 0.9\}$ , 根据 DOE 可产生 16 种参数组合。基于算例 2\_3, 使用 2.6 节 HV 指标评价不同组合下算法对算例的求解性能, 每个组合均运行 10 次取平均值。基于所得结果可得到各参数在不同水平上对 HV 度量的影响趋势, 如图 4 所示。因此, 最佳参数组合为  $\alpha=0.7$ 、 $\gamma=0.8$ 、 $\varepsilon_0=0.6$ 。

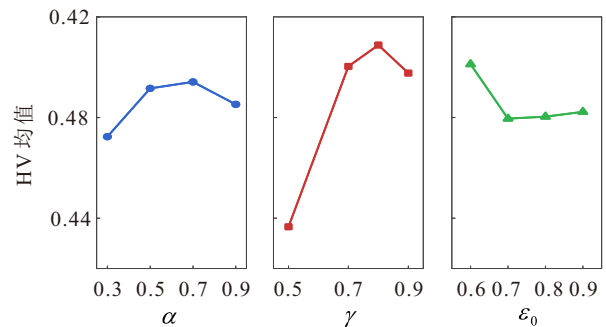


图4 均值主效应图

### 3.3 RINSGA-II 改进策略有效性验证

为验证 RINSGA-II 算法中所提出关键策略的有效性, 在 RINSGA-II 的基础上设计 3 种变体算法, 包括采用随机初始化规则生成初始解的算法 S1; 采用固定交叉变异概率的算法 S2;  $Q$ -learning 模块采用传统  $\varepsilon$ -greedy 策略的算法 S3。通过综合评价指标 IGD 来评价各算法所得 Pareto 解集的收敛性和多样性。每个算例下各算法均独立运行 10 次并取平均值, 结果如表 5 所示, 较优结果用粗体显示。

表5 RINSGA-II 及 3 种变体算法的 IGD 均值

实验	紧急任务	RINSGA-II	S1	S2	S3
1	0	<b>0.0027</b>	0.0204	0.0038	0.0041
	3	<b>0.0497</b>	0.0515	0.0536	0.0509
	5	<b>0.0445</b>	0.0506	0.0524	0.0490
2	0	<b>0.0098</b>	0.1050	0.0163	0.0185
	3	<b>0.0370</b>	0.1132	0.0429	0.0431
	5	<b>0.0457</b>	0.1232	0.0510	0.0539
3	0	<b>0.0109</b>	0.1617	0.0126	0.0124
	3	<b>0.0160</b>	0.1591	0.1447	0.0846
	5	<b>0.0133</b>	0.1766	0.1308	0.0893
4	0	<b>0.0092</b>	0.2416	0.0106	0.0115
	3	0.0369	0.1888	<b>0.0364</b>	0.0405
	5	<b>0.0280</b>	0.2677	0.0287	0.0284
5	0	<b>0.0108</b>	0.2909	0.0124	0.0122
	3	<b>0.0498</b>	0.2812	0.0537	0.0538
	5	<b>0.0395</b>	0.2847	0.0455	0.0442
6	0	<b>0.0088</b>	0.3007	0.0121	0.0100
	3	<b>0.0275</b>	0.3371	0.0583	0.0314
	5	<b>0.0221</b>	0.3882	0.0575	0.0287
7	0	<b>0.0081</b>	0.3666	0.0133	0.0123
	3	<b>0.0163</b>	0.5296	0.0528	0.0515
	5	<b>0.0238</b>	0.5135	0.0518	0.0580
8	0	<b>0.0081</b>	0.3524	0.0104	0.0100
	3	0.0312	0.5421	<b>0.0301</b>	0.0306
	5	<b>0.0293</b>	0.6182	0.0328	0.0307
均值		<b>0.0241</b>	0.2694	0.0423	0.0358

表 5 中, RINSGA-II 在 22 个算例上的 IGD 指标值均为最优; S2 仅在算例 4\_3 和 8\_3 上的 IGD 值表现最优, 另外两个算法未取得最优结果. 这表明所提出的关键策略对 RINSGA-II 的求解性能具有积极影响. 从 IGD 指标的整体均值来看, S2 表现最差, 为 0.2694. 这表明混合初始化策略对算法性能的影响最大, 这是由于该策略可以生成分布性和多样性较好的初始解, 为后续进化提供了更全面的搜索空间, 增加了算法找到前沿解的可能性.

对于基于 Q-learning 的参数调整策略, 为进一步从其作用机制的角度验证该策略的有效性, 以算例 3\_0 为例, 分析了算法在迭代过程中  $P_c$  和  $P_m$  值以及种群 HV 和 Sp 指标值的变化, 相应的变化趋势分别如图 5 和图 6 所示.

图 5 中, 算法在迭代初期,  $P_c$  值会在较高区间波动, 较高的  $P_c$  值能够增强算法的全局搜索能力, 加快算法收敛. 在迭代中后期, 种群逐渐趋于收敛, 种群中个体间差异变小, 此时  $P_c$  值会逐渐下降, 而  $P_m$  值则会随迭代进程缓慢增加, 以避免算法陷入局部最优. 由图 6 可看出, 图 5 中  $P_c$  和  $P_m$  值因迭代的变化特点使得种群的多样性和收敛性不断提升. 上述结果表明, 该策略能够根据进化过程中种群状态选择

最佳动作, 进而引导算法收敛. 因此, 该策略可以有效提升 RINSGA-II 算法的性能.

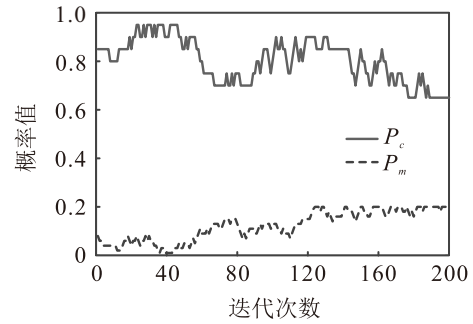


图5  $P_c$ 及 $P_m$ 随迭代过程的变化趋势

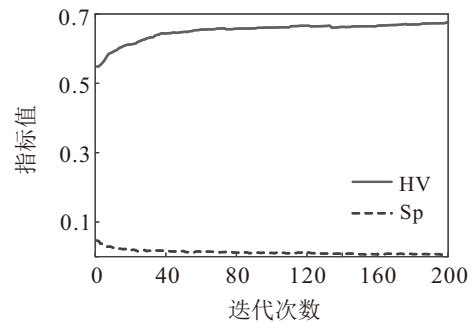


图6 HV和Sp指标值的变化趋势

### 3.4 算法性能对比分析

#### 3.4.1 与其他算法性能对比

分别利用 RINSGA-II 和 4 种对比算法求解上述 24 个问题算例, 每个算例下各算法均独立运行 10 次并取平均值, 两指标的均值如表 6 所示. 特别地, 对于紧急任务不为 0 的算例, 在紧急任务到达时刻之前, 假定各子任务的实际执行时间取标准服务时间, 并基于该假设对模型 M2 求解.

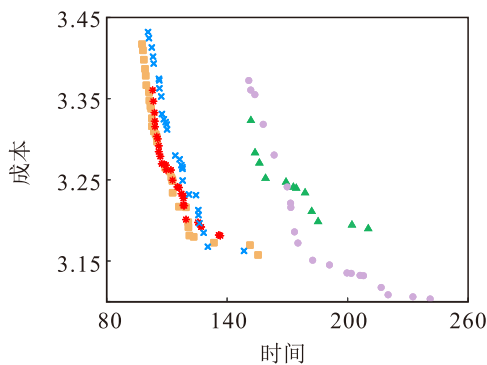
分析表 6 发现: 1) 对于 GD 指标, RINSGA-II 在 17 个算例上取得了最优值; MOEA 在 2 个算例上表现最优; ENSGA-II 在 5 个算例上结果最优. 整体来看, RINSGA-II 在解集收敛性方面优于对比算法. 2) 对于 IGD 指标, RINSGA-II 在 22 算例上的 IGD 均值都明显小于其他算法, 在算例 1\_3 和 1\_5 上的表现仅次于 MOEA 且相差不大, 其余算法均未取得最优结果. 两指标值的结果表明, 与其他算法相比, RINSGA-II 能够得到收敛性和多样性更好的 Pareto 解集.

为直观地比较不同算法求得的非支配解集的分布情况, 分别以算例 4\_0 和 4\_5 为例, 绘制了各算法求解模型 M1 和 M2 的前沿解分布图. 为便于展示, 将各区间目标的目标值用区间众数替代. 从图 7 可以看出, RINSGA-II 求解两个算例得到的前沿解都分布在最外侧; 与另外 4 种算法相比, 本文算法得到的前沿解的多样性最好, 且分布更为均匀.

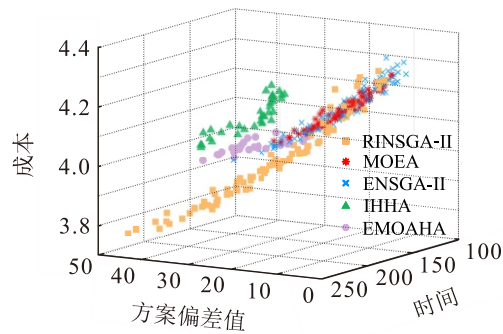
在算法运行效率方面, 图 8 分别绘制了 8 组问

表6 4种算法的GD和IGD指标均值结果

实验	紧急任务	GD					IGD				
		RINSGA-II	MOEA	ENSGA-II	IHHA	EMOAHA	RINSGA-II	MOEA	ENSGA-II	IHHA	EMOAHA
1	0	<b>0.0006</b>	0.0027	0.0009	0.0335	0.0034	<b>0.0028</b>	0.0083	0.0036	0.1036	0.0649
	3	<b>0.0015</b>	0.0017	0.0028	0.0309	0.0039	0.0139	<b>0.0046</b>	0.0182	0.2045	0.0873
	5	<b>0.0014</b>	0.0015	0.0016	0.0395	0.0048	0.0190	<b>0.0099</b>	0.0274	0.2374	0.1051
2	0	<b>0.0013</b>	0.0043	0.0016	0.0733	0.0208	<b>0.0130</b>	0.0444	0.0261	0.1753	0.1525
	3	0.0022	<b>0.0017</b>	0.0025	0.0461	0.0084	<b>0.0256</b>	0.0379	0.0497	0.3753	0.1518
	5	<b>0.0016</b>	0.0021	0.0020	0.0640	0.0108	<b>0.0214</b>	0.0266	0.0448	0.3175	0.1572
3	0	<b>0.0018</b>	0.0034	0.0019	0.0897	0.0546	<b>0.0129</b>	0.1482	0.0947	0.2779	0.2447
	3	<b>0.0023</b>	0.0027	0.0029	0.0718	0.0402	<b>0.0294</b>	0.0714	0.0734	0.4976	0.3001
	5	0.0028	<b>0.0024</b>	0.0032	0.0933	0.0496	<b>0.0336</b>	0.0640	0.0793	0.4688	0.3490
4	0	<b>0.0025</b>	0.0041	0.0026	0.0980	0.0813	<b>0.0189</b>	0.1592	0.1156	0.3564	0.3059
	3	<b>0.0020</b>	0.0022	0.0024	0.0635	0.0239	<b>0.0377</b>	0.1000	0.0904	0.4959	0.3736
	5	<b>0.0017</b>	0.0019	0.0030	0.0645	0.0292	<b>0.0333</b>	0.0952	0.0954	0.5983	0.4084
5	0	0.0023	0.0044	<b>0.0021</b>	0.1244	0.1274	<b>0.0245</b>	0.1704	0.1165	0.3851	0.3269
	3	0.0020	0.0020	<b>0.0018</b>	0.0612	0.0448	<b>0.0337</b>	0.1203	0.1165	0.5535	0.5170
	5	<b>0.0021</b>	0.0022	0.0023	0.0679	0.0424	<b>0.0361</b>	0.1206	0.1144	0.5617	0.4147
6	0	<b>0.0025</b>	0.0034	0.0028	0.1255	0.1198	<b>0.0205</b>	0.2123	0.1539	0.3933	0.3786
	3	0.0023	0.0024	<b>0.0022</b>	0.0656	0.0370	<b>0.0440</b>	0.1372	0.1366	0.5849	0.4552
	5	<b>0.0020</b>	0.0021	0.0027	0.0642	0.0497	<b>0.0396</b>	0.1370	0.1343	0.5680	0.4510
7	0	0.0027	0.0038	<b>0.0021</b>	0.1206	0.0818	<b>0.0188</b>	0.2301	0.2004	0.4932	0.4460
	3	<b>0.0025</b>	0.0026	0.0026	0.0620	0.0558	<b>0.0528</b>	0.1596	0.1386	0.5297	0.4261
	5	0.0025	0.0029	<b>0.0022</b>	0.0511	0.0595	<b>0.0509</b>	0.1446	0.1279	0.6145	0.5113
8	0	<b>0.0025</b>	0.0036	0.0028	0.0898	0.0732	<b>0.0211</b>	0.2072	0.1701	0.4626	0.4087
	3	<b>0.0021</b>	0.0023	0.0022	0.0497	0.0478	<b>0.0446</b>	0.1575	0.1567	0.6089	0.5315
	5	<b>0.0022</b>	0.0024	0.0025	0.0532	0.0474	<b>0.0420</b>	0.1539	0.1564	0.6039	0.4351
均值		<b>0.0021</b>	0.0027	0.0023	0.0710	0.0466	<b>0.0288</b>	0.1133	0.1017	0.4362	0.3334

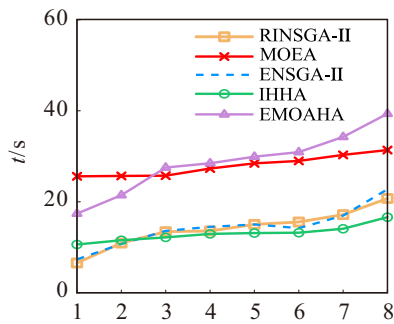


(a) 算例4\_0对应的 Pareto 前沿

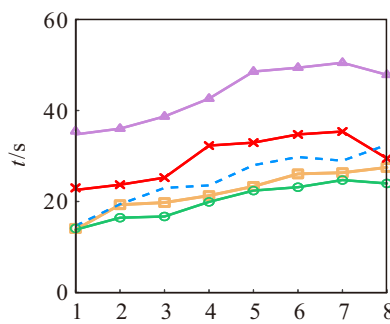


(b) 5种算法的 Pareto 前沿对比

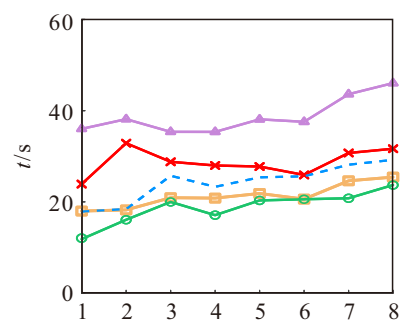
图7 5种算法的 Pareto 前沿对比



(a) R=0



(b) R=3



(c) R=5

图8 5种算法求解不同算例的平均运行时间

题算例在 3 种情形下的算法运行时间趋势图. 可以看出, 在 3 种情形下, 算法效率从高到低分别为 IHHA、RINSGA-II、ENSGA-II、MOEA、EMOHA. RINSGA-II 的运行时间仅略高于 IHHA, 但其所得 Pareto 解集的质量却显著优于 IHHA. 随着算例规模的增大, RINSGA-II 的运行时间增长幅度变小, 表明 RINSGA-II 算法在保证解集质量的同时, 仍能够有效求解大规模问题.

### 3.4.2 不确定参数的敏感性分析

为进一步探讨模型中云服务的服务时间  $\tilde{t}_{ij}^k$  和服务成本  $\tilde{c}_{ij}^k$  参数的不确定性程度对目标值及用户任务

需求约束的影响, 设计不确定参数的敏感性分析实验. 以算例 2\_0 和 2\_5 为例, 在云服务的  $t_{ij}^k$  和  $c_{ij}^k$  标准值基础上, 分别设置不同的波动程度来生成  $\tilde{t}_{ij}^k$  和  $\tilde{c}_{ij}^k$  对应的区间数. 在每个波动水平下分别用 RINSGA-II 求解模型 M1 和 M2, 计算得到 Pareto 解集的平均目标函数值, 并统计用户任务需求约束的满足程度. 约束满足程度可表示为: Pareto 解集中能够满足所有任务的完成时间和成本要求的解所占的比例, 结果如表 7 所示.

分析表 7 可以发现: 1) 对于区间目标  $\tilde{T}$  和  $\tilde{C}$ , 参数波动水平越高, 目标值对应的区间宽度越大. 2) 随

表7 不同波动水平下的模型结果

波动水平	M1			M2			
	$\tilde{T}$	$\tilde{C}$	满足约束解占比	$\tilde{T}$	$\tilde{C}$	DE	满足约束解占比
0% (标准值)	96.30	101357.39	100%	120.07	150327.20	2.73	100%
5% ~ 10%	[87.47,100.57]	[97262.03,112095.96]	82%	[111.61,134.12]	[141103.06,165469.71]	1.94	72%
10% ~ 15%	[89.14,112.71]	[92003.29,113765.62]	70%	[107.99,133.50]	[138615.29,170998.69]	2.47	33%
15% ~ 20%	[80.86,112.58]	[90241.84,120260.14]	41%	[108.81,143.59]	[134770.49,178754.31]	3.10	22%

着参数不确定程度增加, Pareto 解集中完全满足任务需求约束的解所占比例逐渐下降. 但是, 即使在 15% ~ 20% 的波动水平下, Pareto 解集中仍存在足够数量的解能够满足用户需求, 因此仍然能够满足实际生产需要. 3) 从偏差值目标 DE 的结果来看, 当由于紧急任务到达而触发服务重组时, 算法生成的新的服务重组方案与初始服务组合相比, 仅需调整约 3 个子任务的服务选择. 这一结果表明, 本文模型在不同的不确定程度下仍然能够只通过较小的调整应对变化, 体现出了良好的鲁棒性.

综上所述: 本文算法在所求 Pareto 前沿的收敛性和多样性上均显著优于对比算法; 区间多目标服务重组优化模型可以有效地处理 QoS 不确定性和紧急任务扰动, 得到鲁棒性较好的云服务组合方案.

## 4 结论

本文研究了面向多任务的动态云制造 SCO 问题, 综合考虑了云服务 QoS 属性的不确定性和紧急任务扰动. 首先, 采用区间数对云服务的服务时间、服务成本的不确定性进行统一表征, 建立了包含区间目标和约束的多目标云服务重组优化模型. 针对模型特征, 提出了一种 RINSGA-II 算法求解模型. 算法设计了基于启发式规则的混合初始化策略; 提出了基于三角分布下区间数排序可能度计算模型的区间 Pareto 支配关系, 以及区间拥挤距离计算方法; 同时设计了基于 Q-learning 的自适应参数调整策略, 以兼顾种群的多样性并提升算法收敛速度. 最后, 通过 8 组不同规模的问题算例进行了数值实验. 结果

表明, RINSGA-II 中设计的改进策略对算法性能的提升具有积极影响; RINSGA-II 算法在所得 Pareto 解集的收敛性和多样性以及算法运行效率方面均显著优于对比算法; 同时, 本文模型能够有效地吸收 QoS 属性的不确定性并处理紧急任务干扰, 进而得到满足不同用户需求的服务组合方案.

未来将在该研究的基础上, 进一步考虑云制造系统中其他类型扰动的发生, 如云服务动态加入及退出云平台, 进而探索能够同时处理多种扰动的更高效的策略及求解算法.

## 参考文献 (References)

- [1] 李伯虎, 柴旭东, 侯宝存, 等. 云制造系统 3.0 —— 一种“智能+”时代的新智能制造系统[J]. 计算机集成制造系统, 2019, 25(12): 2997-3012.  
(Li B H, Chai X D, Hou B C, et al. Cloud manufacturing system 3.0 — New intelligent manufacturing system in era of “Intelligence+” [J]. Computer Integrated Manufacturing Systems, 2019, 25(12): 2997-3012.)
- [2] Lim M K, Xiong W Q, Lei Z M. Theory, supporting technology and application analysis of cloud manufacturing: A systematic and comprehensive literature review[J]. Industrial Management & Data Systems, 2020, 120(8): 1585-1614.
- [3] Wang T R, Zhang P Z, Liu J, et al. Many-objective cloud manufacturing service selection and scheduling with an evolutionary algorithm based on adaptive environment selection strategy[J]. Applied Soft Computing, 2021, 112: 107737.
- [4] Wan C C, Zheng H L, Guo L, et al. Hierarchical scheduling for multi-composite tasks in cloud manufacturing[J]. International Journal of Production

- Research, 2023, 61(4): 1039-1057.
- [5] Tong H G, Zhu J J. A novel method for customer-oriented scheduling with available manufacturing time windows in cloud manufacturing[J]. *Robotics and Computer-Integrated Manufacturing*, 2022, 75: 102303.
- [6] 李嘉, 杨东. 云制造环境下考虑不确定性的产品配置优化[J]. *系统管理学报*, 2022, 31(2): 384-395.  
(Li J, Yang D. Product configuration optimization problems in cloud manufacturing with uncertainty[J]. *Journal of Systems & Management*, 2022, 31(2): 384-395.)
- [7] 尹璐, 周俊龙, 孙晋, 等. 不确定性感知的边缘计算任务调度算法[J]. *控制与决策*, 2024, 39(7): 2405-2413.  
(Yin L, Zhou J L, Sun J, et al. Uncertainty-aware task scheduling algorithm in edge computing environments[J]. *Control and Decision*, 2024, 39(7): 2405-2413.)
- [8] Xiong W Q, Lim M K, Tseng M L, et al. An effective adaptive adjustment model of task scheduling and resource allocation based on multi-stakeholder interests in cloud manufacturing[J]. *Advanced Engineering Informatics*, 2023, 56: 101937.
- [9] Elgendy A, Yan J H, Zhang M Y. A parallel distributed genetic algorithm using apache spark for flexible scheduling of multitasks in a cloud manufacturing environment[J]. *International Journal of Computer Integrated Manufacturing*, 2024, 37(5): 652-667.
- [10] 王彦凯, 王时龙, 杨波, 等. 一种实际多约束环境下的云制造服务组合动态自适应重构方法[J]. *机械工程学报*, 2023, 59(14): 339-351.  
(Wang Y K, Wang S L, Yang B, et al. A dynamic adaptive reconfiguration method for cloud manufacturing service composition in a practical multi-constraint environment[J]. *Journal of Mechanical Engineering*, 2023, 59(14): 339-351.)
- [11] Dai Z W, Zhang Z Y, Chen M Z. Collaborative task scheduling with new task arrival in cloud manufacturing using improved multi-population biogeography-based optimization[J]. *Journal of Intelligent & Fuzzy Systems*, 2021, 41(2): 3849-3872.
- [12] 董明, 郭书洋. 基于互联网产能共享平台的动态订单分配问题研究[J]. *工业工程与管理*, 2023, 28(1): 199-206.  
(Dong M, Guo S P. Research on dynamic order allocation of Internet-based capacity sharing platform[J]. *Industrial Engineering and Management*, 2023, 28(1): 199-206.)
- [13] Zhang S, Xu Y B, Zhang W Y. Multitask-oriented manufacturing service composition in an uncertain environment using a hyper-heuristic algorithm[J]. *Journal of Manufacturing Systems*, 2021, 60: 138-151.
- [14] She Q P, Wei X C, Nie G H, et al. QoS-aware cloud service composition: A systematic mapping study from the perspective of computational intelligence[J]. *Expert Systems with Applications*, 2019, 138: 112804.
- [15] 韩红桂, 徐子昂, 王晶晶. 基于 Q 学习的多任务多目标粒子群优化算法[J]. *控制与决策*, 2023, 38(11): 3039-3047.  
(Han H G, Xu Z A, Wang J J. A Q-learning-based multi-task multi-objective particle swarm optimization algorithm[J]. *Control and Decision*, 2023, 38(11): 3039-3047.)
- [16] 赵仕存, 周泓. 强化学习驱动进化的模因算法求解准时制分布式柔性作业车间调度问题[J]. *控制与决策*, DOI: 10.13195/j.kzyjc.2025.0310.  
(Zhao S C, Zhou H. Reinforcement learning-driven evolutionary memetic algorithm for solving the just-in-time distributed flexible job shop scheduling problem[J]. *Control and Decision*, DOI: 10.13195/j.kzyjc.2025.0310.)
- [17] 龚日朝, 潘芬萍. 非均匀分布下区间数排序可能度计算模型及其应用[J]. *中国管理科学*, 2020, 28(12): 220-230.  
(Gong R Z, Pan F P. A possibility degree model for ranking interval numbers under non-uniform distribution and its application[J]. *Chinese Journal of Management Science*, 2020, 28(12): 220-230.)
- [18] Gong D W, Qin N N, Sun X Y. Evolutionary algorithms for multi-objective optimization problems with interval parameters[C]. 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications. Changsha, 2010: 411-420.
- [19] Chen M J, Xu J Y, Zhang W Y, et al. A new customer-oriented multi-task scheduling model for cloud manufacturing considering available periods of services using an improved hyper-heuristic algorithm[J]. *Expert Systems with Applications*, 2025, 269: 126419.
- [20] Zhang J X, Guo B, Ding X F, et al. An adaptive multi-objective multi-task scheduling method by hierarchical deep reinforcement learning[J]. *Applied Soft Computing*, 2024, 154: 111342.
- [21] 王春, 王艳, 纪志成. 求解区间柔性作业车间调度的多目标进化算法[J]. *控制与决策*, 2019, 34(5): 908-916.  
(Wang C, Wang Y, Ji Z C. Multi-objective evolutionary algorithm to solve interval flexible job shop scheduling problem[J]. *Control and Decision*, 2019, 34(5): 908-916.)
- [22] Wang Z Q, Liao W Z, Zhang Y P. Rescheduling optimisation of sustainable multi-objective fuzzy flexible job shop under uncertain environment[J]. *International Journal of Production Research*, 2024, 62(24): 8904-8920.
- [23] Zhang Q F, Li S B, Pu R Q, et al. An adaptive robust service composition and optimal selection method for cloud manufacturing based on the enhanced multi-objective artificial hummingbird algorithm[J]. *Expert Systems with Applications*, 2024, 244: 122823.

## 作者简介

张晓冬 (1972-), 女, 教授, 博士生导师, 主要研究方向为制造系统工程、复杂系统仿真, E-mail: xdzhang@manage.ustb.edu.cn;

燕洁晨 (1997-), 女, 博士生, 主要研究方向为生产计划与调度、智能优化算法, E-mail: yjchen0316@163.com;

孙家正 (1994-), 男, 博士生, 主要研究方向为生产管理, E-mail: d202110478@xs.ustb.edu.cn.