

文章编号:1673-9469(2024)05-0103-10

DOI:10.3969/j.issn.1673-9469.2024.05.014

基于改进灰狼算法的云制造服务随机调度优化研究

张广涵,王天日*,薛永梅

(太原理工大学 经济管理学院,山西 太原 030024)

摘要: 为研究云制造背景下分散的制造资源随机调度问题,将服务时间设置为不确定生产参数,建立随机调度模型,并使用与强化学习结合的改进灰狼算法进行求解以优化调度方案。从云平台、用户以及服务提供商三方出发,提出了基于不确定环境的多目标数学模型,同时,将灰狼优化算法与强化学习结合,使用 Q-learning 调整参数,并利用双种群分别进行探索和挖掘,防止陷入局部最优,并用于解决中大规模实例。通过对比实验评估算法性能,该算法在 16 组算例中性能均显示最优,能够提升加工效率并增加相关者收益,对算法参数进行了敏感性分析,验证了参数的合理性和适用性。

关键词: 云制造服务;随机调度;灰狼优化算法;多目标优化;拉丁超立方抽样

中图分类号: TP301.6;TP391.9

文献标识码: A

Research on Random Scheduling Optimization of Cloud Manufacturing Service Based on Improved Grey Wolf Algorithm

ZHANG Guanghan, WANG Tianri*, XUE Yongmei

(School of Economics and Management, Taiyuan University of Technology, Taiyuan, Shanxi 030024, China)

Abstract: To study the random scheduling problem of dispersed manufacturing resources in the context of cloud manufacturing, the service time is set as an uncertain production parameter, a random scheduling model is established, and an improved grey wolf algorithm combined with reinforcement learning is used to solve it to optimize the scheduling scheme. Starting from the perspectives of cloud platforms, users, and service providers, a multi-objective mathematical model based on uncertain environments is proposed. At the same time, the Grey Wolf Algorithm is combined with reinforcement learning, Q-learning is used to adjust parameters, and dual populations are explored and mined separately to prevent falling into local optima and to solve medium to large-scale instances. By conducting comparative experiments to evaluate the performance of the algorithm, it was found that the algorithm showed the best performance in all 16 cases, which can improve processing efficiency and increase stakeholder benefits. Sensitivity analysis was also conducted on the algorithm parameters to verify their rationality and applicability.

Key words: cloud manufacturing services; random scheduling; grey wolf optimizer; multi-objective optimization; Latin hypercube sampling

随着新时代信息技术的诞生,智能制造已成为全球技术创新和经济竞争的重点^[1],企业面临着市场需求变化迅速、产品复杂性提升^[2]的挑战,而单一的制造企业很难有效完成复杂的制造任务

和满足个性化的用户与市场需求^[3]。在这种情况下,拥有较高的生产利润、较低的管理风险的分布式制造模式逐渐被企业所重视,进而成为制造企业的普遍趋势^[4]。在这样的背景下,伴随云计

收稿日期:2024-03-10

基金项目:山西省基础研究计划项目(202303021221035);山西省回国留学人员科研资助项目(2023-070);国家自然科学基金资助项目(71701141)

作者简介:张广涵(1999-),女,河北石家庄人,硕士研究生,从事多目标算法优化、强化学习、云制造服务调度方面的研究。

* 通信作者:王天日(1986-),男,山西大同人,博士,教授,从事运筹与优化、能源系统工程管理方面的研究。

算^[5]、物联网^[6] (IoT)、边缘计算和雾计算^[7]等先进信息技术在制造领域应用不断成熟,云制造(CMfg)模式下的分布式调度优化研究逐渐成为许多学者与专家研究的热点内容。

过去研究大多集中在确定情况下的优化方案,Lu 等人^[8]研究分布式混合流程调度问题,设计了一种多目标混合迭代贪婪算法,降低完工时间并减少了能耗。Wang 等人^[9]考虑到分布式多目标优化的复杂性,提出一种协同算法来求解分布式流水车间的节能调度问题。现实中,子任务完成时间不确定、订单交付时间延迟等情况多有发生,张安等^[10]考虑无人机巡航速度等不确定因素,基于模糊可信性理论构建模糊机会约束规划模型,并提出多策略融合的灰狼优化算法优化解决任务分配问题。Ghamginzadeh 等^[11]通过模糊时间研究多目标多技能项目调度问题,并提出一种多目标帝国竞争算法进行求解。于冬梅等^[12]考虑应急设施的共享不确定因素,将带混沌搜索策略的改进灰狼优化算法应用于选址决策问题中,并验证了算法优势。综上,确定性调度方法在现实中可能导致不可执行的方案,因此,考虑在不确定环境下的多任务制造服务组合调度优化十分有必要。为获得更优的决策方案,本文从用户、云平台和服务制造商的多角度出发,考虑任务加工时间的不确定性,建立不确定环境下的多目标任务调度模型,并通过改进算法求解模型,提升调度方案质量,协同优化多方利益方需求,为企业在风险下作出最优决策提供有效参考。

1 问题描述与模型

1.1 问题描述

本文提出 CMfg 系统中基于用户、云平台以及服务商三方的多目标模型,图 1 为云制造模式下调度框架图。假设 N 个用户向共享平台提交多个生产任务,用户记为 $U = \{U_1, U_2, \dots, U_l, \dots, U_n\}$,第 i 个用户有 J_i 个订单任务,所有任务都拆分为多个子任务操作,第 j 个任务有 L_j 个子任务, st_{ijl} 表示第 i 个用户第 j 个任务的第 l 个子任务,这些子任务均可由云服务平台注册的 M 家服务提供商所执行,每家服务商拥有数量 K_m 的机器,其中 $k \in 1, 2, \dots, K_m$,其中 E_{mk} 为服务商 m 中的机器 k 。

对于所设问题提出以下假设:(1)用户的任务彼此独立,具有相同的优先级,并且每个任务可细分为多个子任务;(2)每个云服务能够执行多种类型的子任务,但同时只能处理一个子任务;(3)每个子任务有多个可选的云服务,但只能由一个云服务执行,并且执行过程不可中断;(4)按照用户提交时间和交货期的要求,将所有任务开始时刻转化为零;(5)每个子任务在不同的云服务上具有不同的服务时间和成本,且这些信息提前已知;(6)在调度前完成任务分解和可行的服务组合;(7)每个工厂拥有许多不同能力的机器,这些机器可以满足对应订单的技术要求;(8)机器处于不同状态时,能耗也不同;(9)平台对每个工厂有固定的运输时间,超过需要支付额外的延迟成本。

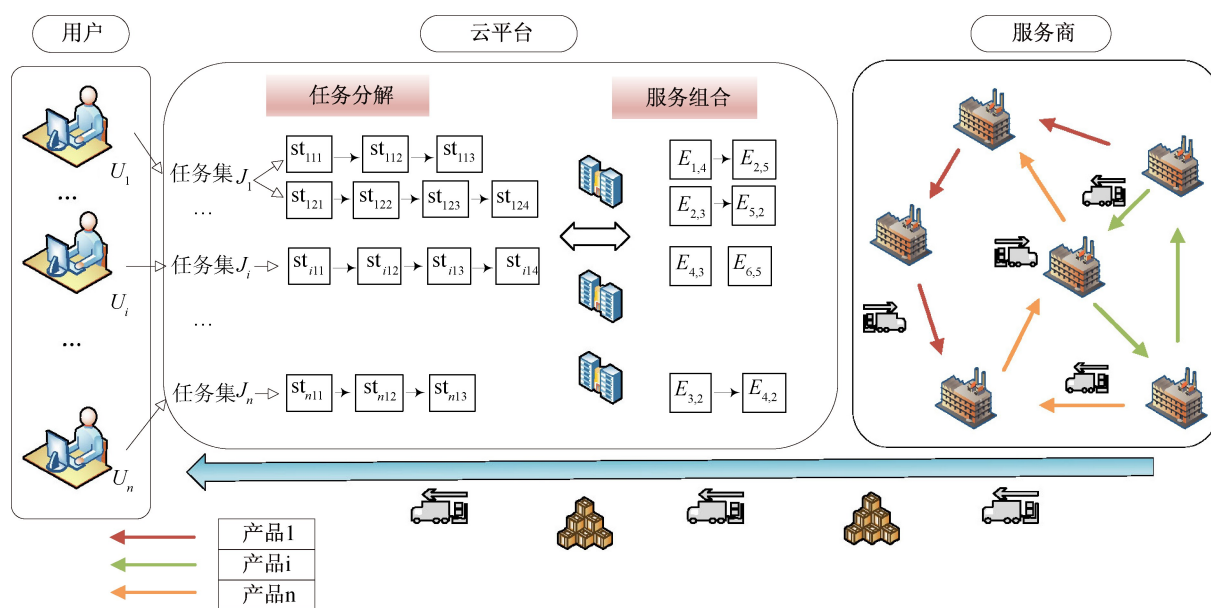


图 1 云制造调度模型框架

Fig. 1 Framework of cloud manufacturing scheduling model

1.2 模型建立

对于不同的利益方,需求也不同,本文将用户完工时间 f_1 ,云平台收益 f_2 以及服务商利润 f_3 作为求解的目标。其中 X_{ijl}^{mk} 、 $Y_{ijl,abp}^{mk}$ 为决策变量,如果 st_{ijl} 在 E_{mk} 上运行则 X_{ijl}^{mk} 为1,否则为0;同一个机器上加工的两个操作加工时间不能重叠, st_{ijl} 顺序在 st_{abp} 之前则 $Y_{ijl,abp}^{mk}$ 为1,否则为0。目标函数及模型约束描述如下:

1.2.1 用户最小完工时间

完工时间由所有任务最大完工时间 T 决定, ST_{ijl} 表示 st_{ijl} 开始时间, $\widetilde{OT}_{ijl}^{mk}$ 为对应子任务所需服务时间,是不确定因素, $D_{mm'}$ 表示服务提供商 m 和 m' 间的距离, lt 是单位距离的平均物流成本。用户最小完工时间可由式(1)表示:

$$\begin{aligned} \min f_1 &= \max T_i \\ \min f_1 &= \max \left(\sum_{j=1}^{J_i} \sum_{l=1}^{L_j} (ST_{ijl} + \sum_{m=1}^M \sum_{m'=1}^M \sum_{k=1}^{K_m} \sum_{k'=1}^{K_m} X_{ijl}^{mk} \times \right. \\ &\quad \left. X_{ij(l+1)}^{m'k'} \times (\widetilde{OT}_{ijl}^{mk} + D_{mm'} \times lt) \right) \end{aligned} \quad (1)$$

1.2.2 云平台收益

云平台需考虑到订单抽成收益、运输成本以及延迟成本三部分,其中 ϑ 为每笔订单的抽成比例, PPO_{ijl}^{mk} 为对应订单价格, lc 是单位距离的平均物流时间, FT_{ijl_j} 为任务完成时间,由于加工时间的波动,因此完工时间也未固定,存在提前与延迟交付订单的情况,存在订单延迟成本, TC 为单位时间的延迟成本。云平台收益可由式(2)表示:

$$\begin{aligned} \max f_2 &= \sum_{i=1}^I \left\{ \sum_{j=1}^{J_i} \sum_{l=1}^{L_j-1} \sum_{m=1}^M \sum_{k=1}^{K_m} [\vartheta \times PPO_{ijl}^{mk} \times X_{ijl}^{mk} - \right. \\ &\quad \left. \sum_{m'=1}^M \sum_{k'=1}^{K_m} D_{mm'} \times lc \times X_{ij(l+1)}^{m'k'} \times X_{ijl}^{mk}] - \right. \\ &\quad \left. \sum_{j=1}^{J_i} \sum_{m=1}^M \sum_{k=1}^{K_m} X_{ijl_j}^{mk} \times (\widetilde{FT}_{ijl_j}^{mk} - FT_{ijl_j}) \times TC \right\} \end{aligned} \quad (2)$$

1.2.3 服务商利润

除了订单所获收益外,服务商还需考虑机器加工成本与能耗成本, OC_{ijl}^{mk} 为 st_{ijl} 在 E_{mk} 上的服务成本,机器能耗成本 EC 分为机器加工时能耗 EC_r 与空闲时能耗 EC_d 两种情况, EC_r 计算只需考虑每个在该机器上加工的任务加工时间之和即可, P_{run}^k 为机器 k 运行时的单位能耗成本, EC_d 中计算机器空闲时间需得到开机与关机时间,开机时间即为机器加工第一项子任务的初始时刻,关机时间则为最大完工时间, P_{idle}^k 为机器 k 空闲时的单位能耗成本。服务商利润可由式(3)表示:

$$\max f_3 = \sum_{i=1}^I \sum_{j=1}^{J_i} \sum_{l=1}^{L_j} \sum_{m=1}^M \sum_{k=1}^{K_m} (PPO_{ijl}^{mk} \times X_{ijl}^{mk} - OC_{ijl}^{mk} \times X_{ijl}^{mk} - EC) \quad (3)$$

$$EC = EC_d + EC_r \quad (4)$$

$$EC_r = \sum_{i=1}^I \sum_{m=1}^M \sum_{k=1}^{K_m} \sum_{j=1}^{J_i} P_{run}^k \times \widetilde{OT}_{ijl}^{mk} \times X_{ijl}^{mk} \quad (5)$$

$$EC_d = \sum_{m=1}^M \sum_{k=1}^{K_m} P_{idle}^k \times$$

$$\left(\max T - \sum_{i=1}^I \sum_{j=1}^{J_i} \sum_{l=1}^{L_j} (\widetilde{OT}_{ijl}^{mk} \times X_{ijl}^{mk} - \min ST_{ijl}^{mk}) \right) \quad (6)$$

1.2.4 模型约束

$$\sum_{j=1}^{J_i} \sum_{l=1}^{L_j} ((1 + \vartheta) PPO_{ijl}^{mk}) \leq UP_i \quad (7)$$

$$\sum_{m=1}^M \sum_{k=1}^{K_m} X_{ijl}^{mk} = 1, i \in I, j \in J_i, l \in L_j \quad (8)$$

$$\sum_{j=1}^{J_i} \sum_{l=1}^{L_j} X_{ijl}^{mk} \leq 1, i \in I, m \in M, k \in K_m \quad (9)$$

$$\begin{aligned} &ST_{ij(l-1)}^{mk} + \sum_{m=1}^M \sum_{k=1}^{K_m} X_{ij(l-1)}^{mk} \times \\ &(\widetilde{OT}_{ijl}^{mk} + \sum_{m'=1}^M \sum_{k'=1}^{K_m} \times lt \times X_{ijl}^{m'k'}) \leq ST_{ijl} \\ &i \in I, j \in J_i, l-1 \in L_j \end{aligned} \quad (10)$$

$$\begin{aligned} ST_{ijl}^{mk} &\geq \max(\widetilde{FT}_{ij(l-1)}^{m'k'} + \sum_{m'=1}^M \sum_{k'=1}^{K_m} X_{ij(l-1)}^{m'k'} \times \\ &X_{ijl}^{mk} \times D_{mm'} \times lt, \widetilde{FT}_{abp}^{mk}) \\ &i, i' \in I, j, j' \in J_i, l, l' \in L_j \\ &(i \neq i' \parallel i=i', j \neq j'), m \in M, k \in K_m \end{aligned} \quad (11)$$

$$\begin{aligned} Y_{ijl, i'j'l'}^{mk} + Y_{i'j'l', ij}^{mk} &\leq 1, i, i' \in I, j, j' \in J_i, l, \\ l' \in L_j (i \neq i' \parallel i=i', j \neq j'), m \in M, k \in K_m \end{aligned} \quad (12)$$

$$\begin{aligned} Y_{ijl, l'j'l'}^m + Y_{i'j'l', ij}^m &\geq X_{ijl}^{mk} + X_{i'j'l'}^{mk} - 1, i, i' \in I, j, j' \in J_i, \\ l, l' \in L_j (i \neq i' \parallel i=i', j \neq j'), m \in M, k \in K_m \end{aligned} \quad (13)$$

其中约束(7)表示用户交付金额符合预期, UP_i 为用户预期订单最大可支付金额;约束(8)和约束(9)表明每个工序只能在一台机器上进行加工,且每台机器一次只能加工一道工序;约束(10)规定了同一任务的子任务必须按照预先定义的加工顺序进行,即某个子任务的前序子任务必须全部完成后,该子任务才能开始加工;约束(11)确保同一云服务上的两个子任务的加工时间不会重叠;约束(12)和(13)表示在同一服务上处理不同

子任务时,二进制变量 $Y_{ijl,i'j'l'}^{mk} + Y_{i'j'l',ijl}^{mk} = 1$ 。

2 改进灰狼算法设计

任务调度问题是一个典型的 NP-hard 问题^[13],为了获得更优的决策解,本文采用离散的灰狼优化算法。由于参数少、收敛速度快且具有较好的全局搜索能力。而原始的灰狼优化算法(GWO)在求解连续问题时其求解精度与稳定性方面有较大优势,对于离散问题的求解并不能直接应用。因此,本文基于 GWO 的全局搜索性能和 Q-learning 的反复探索和学习能力,在 GWO 算法框架上进行算法改进、提高局部搜索能力,求解最优策略,从而设计了改进灰狼算法(GWOQL)算法,

该算法能增加解的多样性,提高解的质量。GWOQL 的算法流程如图 2 所示。

2.1 随机抽样

本文主要考虑任务加工时间作为不确定性因素的参数,依据文献[14]本研究采用拉丁超立方抽样^[15]作为抽样方法,假设参数均匀波动,在参数本身[0.8,1.2]的波动范围内进行 1 000 次采样。为了采用这些客观值来评估解决方案的性能,采用平均有效目标函数值机制^[14,16]计算目标值。由于该优化模型中考虑分布式离散背景,目标函数的积分值难以计算,因此,根据大数定律,本研究假设样本的平均目标值等于理论均值有效目标的期望值。

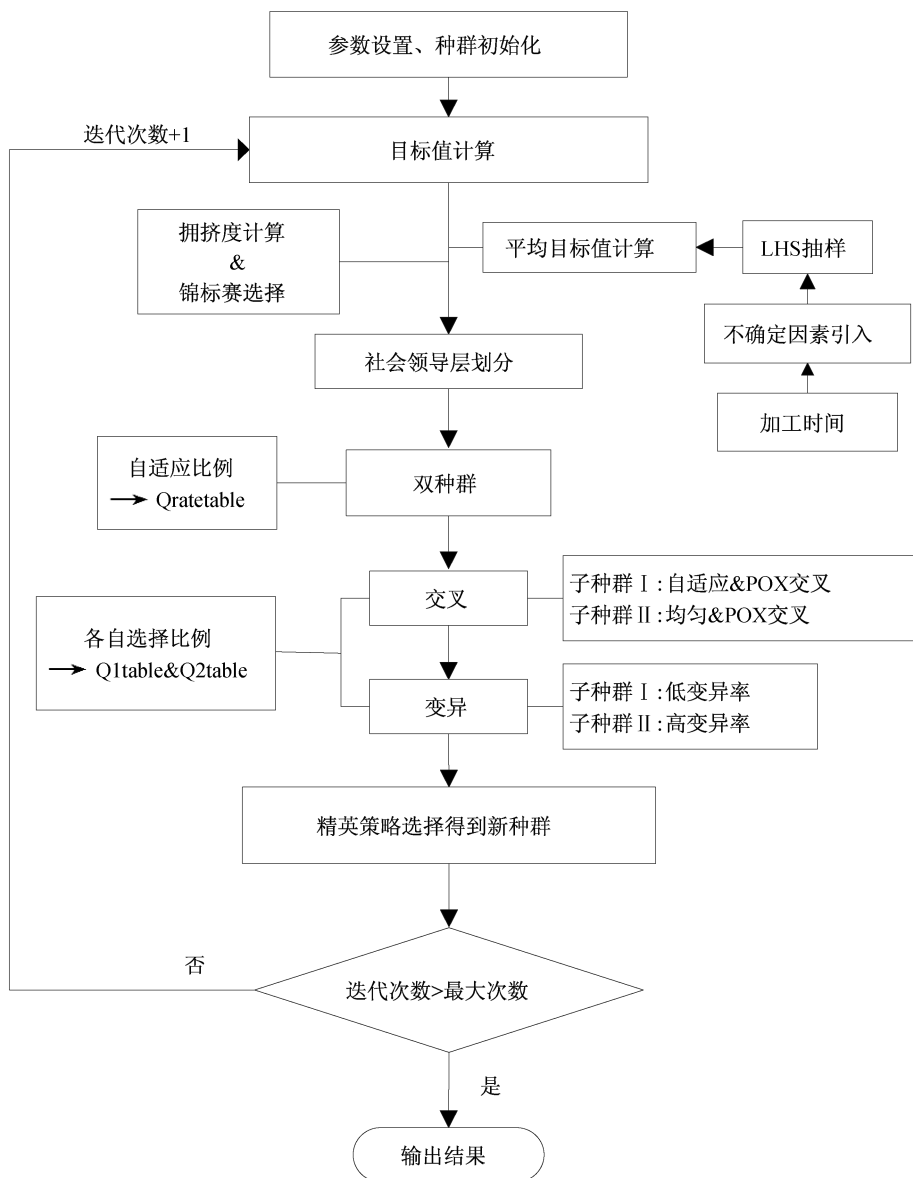


图2 算法框架流程

Fig. 2 Algorithm framework process

2.2 算法步骤

根据问题特性,本文采用二元向量实数编码方式对决策变量进行编码,染色体分为两个向量:第一个向量表示子任务选择的云服务,第二个向量表示子任务的排序。

2.2.1 社会领导层级机制

GWO 的核心思想是找到种群中三个适应值最好的灰狼 (α, β, δ), 其他个体 (ω) 利用这些个体的引导对整个解空间进行探索, 加快种群的收敛。在多目标优化问题中, 根据领导个体筛选规则^[16]将种群划分为多个非支配层, 确定 α, β, δ 个体:

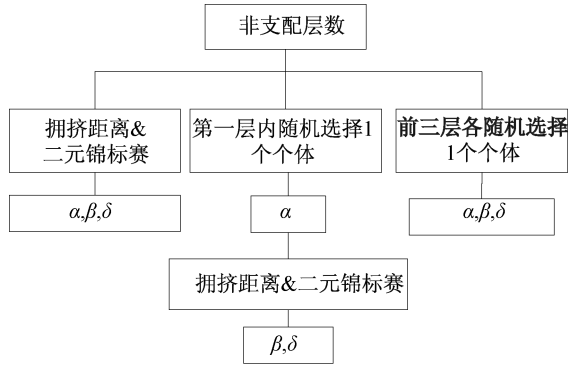


图3 社会领导层级机制

Fig. 3 Social leadership hierarchy mechanism

2.2.2 捕食行为机制

为了保持种群有良好的搜寻方向, 使每只 ω 狼可根据在当前种群中的非支配等级选择领导个体^[16]。对于 GWO, 提高种群算法性能的关键在于平衡探索与开发之间的关系, 早期需要较高的探索能力来加速收敛, 而后期迭代过程中则需加强开发, 以提高解决方案的质量。因此, 本文通过建立双种群实现探索与开发间的平衡, 将整个种群分为两个子种群。通过设计不同的控制参数和交叉策略, 将两个子种群的重点分别放在勘探和开发上。为了加速整个种群接近最优解决方案, 两个种群随机互相交换个体, 以加强种群间信息的交流合作。

两个子种群的染色体分别应用两种不同的交叉操作, 子任务排序部分均采用 POX 交叉^[17]。为了增加种群的多样性, 云服务选择部分子种群 I 被用于开发种群, 采用自适应交叉机制^[18], 子种群 II 则选采用均匀交叉机制^[19], 并定义一个个体 G 的引导概率 pf , 有利于防止算法陷入局部最优^[16]。

利用变异算子的局部搜索能力可以加快收敛

进程, 保持种群的多样性, 并且防止过早的成熟收敛。基于问题特征, 对于子种群 I 采用低突变概率, 子种群 II 则采取高突变概率, 如公式 (14) 所示。本文采用交换的变异算子, 对于云服务选择部分, 使用单点变异, 在子任务对应云服务序列中随机选择一个并进行替换; 对于子任务排序则随机选取两个基因位并进行交换, 以确保生成不会破坏解的可行性的后代。

$$P_m = \begin{cases} 0.25 - 0.2 \times \text{rand} \times \frac{t}{\max It}, & \text{子种群 I} \\ 0.85 - 0.55 \times \text{rand} \times \frac{t}{\max It}, & \text{子种群 II} \end{cases} \quad (14)$$

2.2.3 Q-learning 调参过程

Q-learning 是目前最流行的强化学习算法之一, 算法中包含 Agent、Environment、State、Action 以及 Reward 五部分, 它与环境交互时基于奖励或惩罚来评估其结果, 并使用 Q 表来记录状态和动作之间的关系。

通过调整双种群比例、交叉概率两类参数以优化算法性能与结果。在调整种群比例时, Q-learning 的奖励值对象为两个子种群, 而在进行交叉概率调整过程中, 奖励值所对应的对象分别为种群中的单个个体, 因此, 对于不同的参数调整, 需要引用不同的奖励值机制实现 Q 表的更新。具体详见算法 1。

$$\text{Reward} = 1 - (\text{IGD I} + \text{IGD II}) \quad (15)$$

$$\text{Reward}(\text{Cross I}) = \sum_{i=1}^{N_I} \frac{p_i - o_i}{p_i} \quad (16)$$

$$\text{Reward}(\text{Cross II}) = \sum_{i=1}^{N_{II}} \frac{p_i - o_i}{p_i} \quad (17)$$

算法 1 Q-learning 的调参伪代码:

- (a) 种群初始化;
- (b) 使用 2.1 节中描述方法进行参数抽样与目标值计算;
- (c) 按照 2.2 中内容对种群进行领导层划分;
- (d) 获取种群与个体的 State、Action (初始状态均为 1);
- (e) 通过 Q-learning 获取 Action (种群、个体) 参数选择;
- (f) 根据 Action (种群) 进行双种群划分;
- (g) 双种群分别按照各自 Action (个体) 进行交叉选择, 变异选择按照公式 (14);
- (h) 按照公式 (15) (16) (17) 记录种群与个体

的奖励值;

(i)更新 Q 表;

(j)记录最后的 State、Action(种群、个体)并存储;

(k)判断是否满足最大迭代次数,是则跳回(d),否则输出更新种群信息。

3 结果分析

3.1 实验设置

根据正交实验设计,生成问题组进行测试,如表 1 所示。对于每一组数据,根据在一家生产水泥设备的企业中的调查,在一定范围内随机生成数据,单位时间的子任务延误成本为 0.02 元/h,如表 2 所示。

表 1 正交实验参数

Tab. 1 Orthogonal experimental parameters

测试组	用户数	订单数	种类	服务商数
1	3	10	10	10
2	3	20	15	20
3	3	30	20	30
4	3	40	30	50
5	5	10	15	10
6	5	20	10	20
7	5	30	30	30
8	5	40	20	50
9	7	10	20	10
10	7	20	30	20
11	7	30	15	30
12	7	40	10	50
13	9	10	30	10
14	9	20	20	20
15	9	30	10	30
16	9	40	15	50

表 2 数据参数

Tab. 2 Parameters

参数	取值
单位时间 $lt/(h \cdot km^{-1})$	0.004
单位成本 $lc/(元 \cdot km^{-1})$	0.005
两地距离 D/km	[200, 500]
服务时间 OT/h	[5, 20]
服务成本 $OC/元$	[10, 30]
订价 $PPO/元$	[80 000, 200 000]
服务空闲成本 $P_{idle}/(元 \cdot h^{-1})$	0.7
服务运行成本 $P_{run}/(元 \cdot h^{-1})$	0.9
订单抽成比例 ϑ	0.08

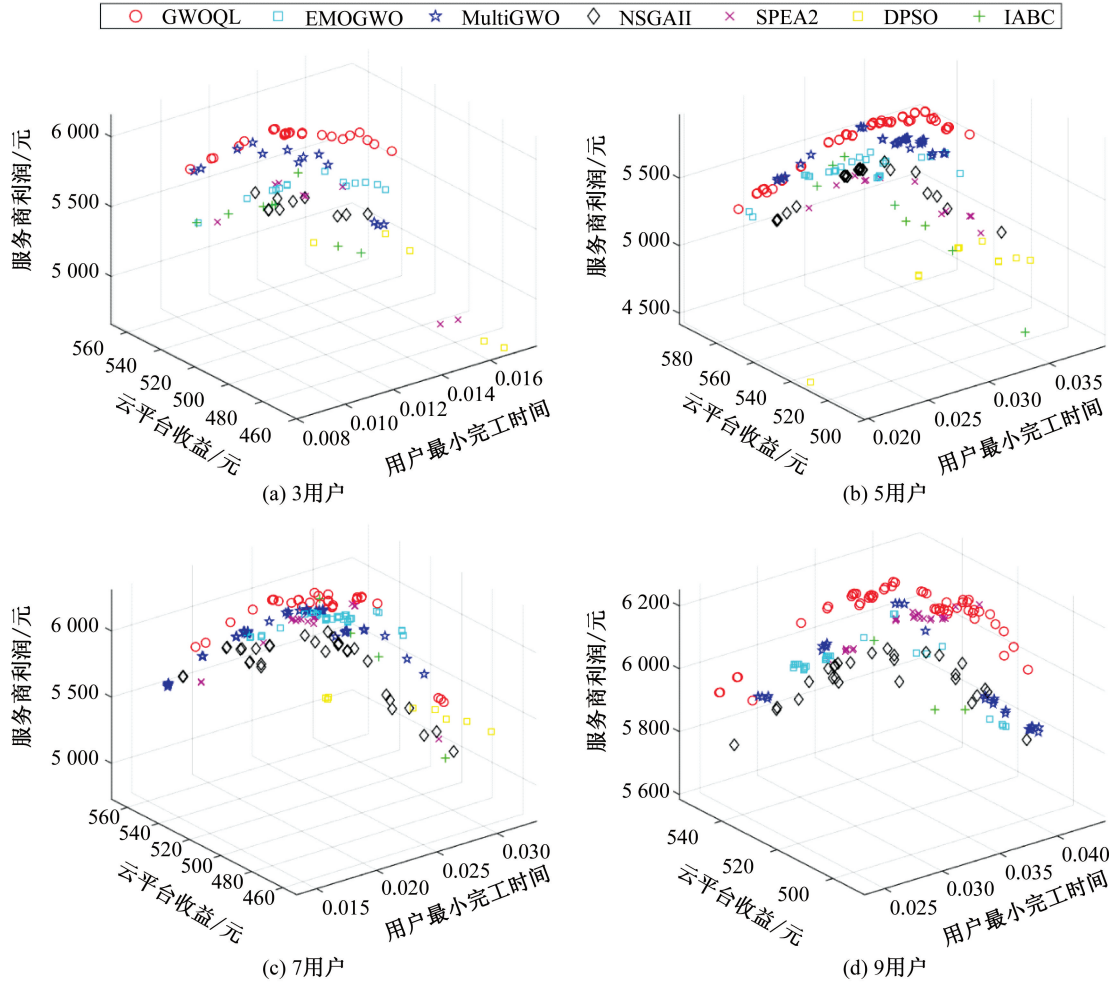
3.2 实验结果及分析

在本节中,设计了详细的实验来评估 GWOQL 的性能。GWOQL 和比较算法是在 Windows10 下采用规格为 AMD Ryzen 7 5800 H with Radeon Graphics@3.20 GHz 与 16 G RAM 的设备在 MATLAB2021a 上进行编码。为了验证本文提出的混合算法的有效性,将该算法与五种求解相似问题的多目标进化算法 NSGA-II、SPEA2、EMOGWO、DPSO-SA、IABC 以及双种群下的 GWO (Multi-GWO) 进行比较,参考已有文献设置^[9,16],所有算法在每个实例上独立运行 20 次,种群数量为 100,迭代次数设置为 200,需要注意的是,为了验证该算法的收敛性和多样性,经过 20 次独立运行后,收集平均结果进行性能比较, pf_{max} 、 pf_{min} 分别为 0.5、0.4,其他算法参数按原始文献中内容进行设置。

每个操作可选择的气器的个数为所有云服务器机器总数的 40%~80%。在算法中考虑了三个目标,将 GWOQL 与其他算法均独立运行 20 次,并进行比较,最后将所有算法的结果汇总所求得的非支配解集作为真实帕累托解集。GWOQL 算法采用了有效的交换个体优化策略和双种群搜索空间探索方法,使得算法能够在局部和全局搜索之间更好地平衡、更快地收敛,以得到最优解;通过 Q-learning 的参数调整,过程中进行不断学习,选取合适参数,使算法更好地适应问题的特性,从而提高了算法的性能。

图 4 为算例中对应 3 名用户共待加工 10 项任务、5 名用户共待加工 20 项任务、7 名用户共待加工 30 项任务与 9 名用户共待加工 50 项任务的各算法帕累托结果的分布情况,从图中可以看出,七种算法中 GWOQL 解的支配性强于其他算法,并且分布更加均匀。GWOQL 在四组试验中都产生了高质量的解,说明该算法能产生更稳定、更有效的解,性能更优,能够在有效缩短用户完工时间的同时,协同增加云平台收益与服务商利润,为制造过程中的多个利益方共同提升利益,实现未来可持续发展。另外,该算法在规模增大时仍能够保持较好性能,说明 GWOQL 有效地利用了并行计算资源,具有较好的并行性和扩展性。

表 3 和表 4 分别列出了所有算法的 GD 和 HV 指标的平均值、指标排名及与最优值对比近似(=)、差于(+)结果。GD 衡量算法的收敛性能,HV 衡量算法的综合性能,最优结果用灰色背景突出显示。Wilcoxon 秩和检验的结果列在最后一行,



注:用户最小完工时间数值为倒数形式,无单位。

图 4 算法结果比较

Fig. 4 Comparison of algorithm results

表 3 GD 比较结果

Tab. 3 GD comparative results

	GWOQL	EMOGWO	MultiGWO	NSGAII	SPEA2	DPSO	IABC	用户/订单/ 种类/服务商
1	0.050 8[1]	0.085 5[3]+	0.078 3[2]+	0.196 6[4]+	0.204[5]+	0.453 1[7]+	0.311[6]+	3/10/10/10
2	0.038 8[1]	0.085 8[3]+	0.067 0[2]+	0.273 9[5]+	0.260 7[4]+	0.585 6[7]+	0.282 8[6]+	3/20/15/20
3	0.058 6[1]	0.067 8[3]=	0.064 7[2]+	0.306 3[5]+	0.265 8[4]+	0.429 7[7]+	0.300 3[6]+	3/30/20/30
4	0.041 0[1]	0.106 8[3]+	0.082 2[2]+	0.162 0[5]+	0.170 3[6]+	0.121 3[4]+	0.174 1[7]+	3/40/30/50
5	0.014 5[1]	0.060 0[3]+	0.041 2[2]+	0.161 9[5]+	0.171 3[6]+	0.100 2[4]+	0.191 2[7]+	5/10/15/20
6	0.013 1[1]	0.045 3[3]+	0.035 4[2]+	0.108 4[4]+	0.121 3[5]+	0.346 8[7]+	0.186 9[6]+	5/20/10/30
7	0.035 2[1]	0.054 8[2]+	0.060 5[3]+	0.163 7[4]+	0.176 1[5]+	0.318 0[7]+	0.200 4[6]+	5/30/30/50
8	0.048 6[1]	0.093 5[3]+	0.062 6[2]+	0.246 1[6]+	0.212 1[4]+	0.280 4[7]+	0.244 4[5]+	5/40/20/10
9	0.006 1[1]	0.016 3[4]+	0.011 3[2]=	0.042 2[5]+	0.045 5[7]+	0.018 1[3]+	0.043 7[6]+	7/10/20/30
10	0.048 8[1]	0.171 4[3]+	0.101 8[2]+	0.450 1[6]+	0.448 5[5]+	0.635 9[7]+	0.383 4[4]+	7/20/30/50
11	0.024 5[1]	0.038 0[2]+	0.040 6[3]+	0.077 8[5]+	0.066 5[4]+	0.149 5[7]+	0.097 5[6]+	7/30/10/10
12	0.024 0[1]	0.038 5[2]+	0.053 3[3]+	0.164 8[6]+	0.130 8[5]+	0.298 3[7]+	0.094 6[4]+	7/40/15/20
13	0.039 8[1]	0.083 0[3]+	0.047 4[2]+	0.292 4[6]+	0.277 6[5]+	0.390 1[7]+	0.250 6[4]+	9/10/30/50
14	0.043 4[1]	0.066 1[2]+	0.076 9[3]+	0.164 6[4]+	0.180 7[6]+	0.169 3[5]+	0.211 7[7]+	9/20/20/10
15	0.028 6[1]	0.074 3[2]+	0.089 7[3]+	0.310 9[5]+	0.328 3[6]+	0.518 8[7]+	0.309 1[4]+	9/30/15/20
16	0.024 2[1]	0.038 5[2]+	0.046 3[4]+	0.060 3[5]+	0.046 1[3]+	0.520 3[7]+	0.174 4[6]+	9/40/10/30
	b/s/w	15/1/0	15/1/0	16/0/0	16/0/0	16/0/0	16/0/0	

注:b/s/w 表示该算法优于、类似、差于其他 6 种算法。

表 4 HV 比较结果

Tab. 4 HV comparative results

	GWOQL	EMOGWO	MultiGWO	NSGAI	SPEA2	DPSO	IABC	用户/订单/ 种类/服务商
1	0.206 5[1]	0.206 1[2] =	0.165 5[5] +	0.178 1[3] +	0.166 6[4] +	0.033 7[7] +	0.073 1[6] +	3/10/10/10
2	0.355 0[1]	0.158 7[2] +	0.107 2[6] +	0.140 7[5] +	0.146 6[3] +	0.020 6[7] +	0.135 4[4] +	3/20/15/20
3	0.296 3[1]	0.060 2[6] +	0.113 3[3] +	0.069 1[5] +	0.100 7[4] +	0.011 9[7] +	0.123 3[2] +	3/30/20/30
4	0.726 8[1]	0.502 9[3] +	0.549 0[2] +	0.093 1[6] +	0.095 8[5] +	0.034 3[7] +	0.223 2[4] +	3/40/30/50
5	0.681 2[1]	0.434 8[3] +	0.477 1[2] +	0.110 2[6] +	0.119 7[5] +	0.039 7[7] +	0.184 3[4] +	5/10/15/20
6	0.510 6[1]	0.271 0[2] +	0.267 4[3] +	0.172 6[5] +	0.174 2[4] +	0.036 3[7] +	0.137 4[6] +	5/20/10/30
7	0.512 8[1]	0.259 7[2] +	0.216 6[3] +	0.090 6[5] +	0.077 4[6] +	0.012 0[7] +	0.152 7[4] +	5/30/30/50
8	0.637 3[1]	0.414 5[3] +	0.549 0[2] +	0.146 9[5] +	0.145 8[6] +	0.068 2[7] +	0.232 5[4] +	5/40/20/10
9	0.087 4[1]	0.088 4[3] =	0.087 5[2] =	0.062 2[7] +	0.063 3[6] +	0.064 3[5] +	0.072 1[4] =	7/10/20/30
10	0.231 0[1]	0.073 2[4] +	0.103 3[3] +	0.067 6[6] +	0.070 8[5] +	0.014 3[7] +	0.130 3[2] +	7/20/30/50
11	0.615 4[1]	0.500 0[2] +	0.435 3[3] +	0.362 9[6] +	0.413 2[4] +	0.059 0[7] +	0.414 1[5] +	7/30/15/10
12	0.344 0[1]	0.133 6[3] +	0.117 8[6] +	0.152 9[4] +	0.128 7[5] +	0.010 9[7] +	0.136 7[2] +	7/40/10/20
13	0.388 0[1]	0.071 4[5] +	0.115 3[2] +	0.097 6[3] +	0.087 1[4] +	0.017 3[7] +	0.069 6[6] +	9/10/30/50
14	0.586 9[1]	0.383 5[2] +	0.335 1[3] +	0.132 4[6] +	0.144 8[5] +	0.040 5[7] +	0.227 9[4] +	9/20/20/10
15	0.449 4[1]	0.130 0[4] +	0.191 3[2] +	0.077 7[6] +	0.117 8[5] +	0.018 7[7] +	0.144 7[3] +	9/30/10/20
16	0.482 8[1]	0.295 9[4] +	0.338 8[3] +	0.283 0[5] +	0.372 3[2] +	0.023 3[7] +	0.149 4[6] +	9/40/15/30
	b/s/w	14/2/0	15/1/0	16/0/0	16/0/0	16/0/0	15/1/0	

b/s/w 表示该算法优于、类似、差于其他 6 种算法。结果显示,GWOQL 在两种指标下均表现最好,在 Wilcoxon 秩和检验方面,在所有 GD 值中 EMOGWO 与 MultiGWO 分别有 1 组类似于 GWOQL,HV 中则分别有 2 组与 1 组近似,其余组别 GWOQL 均明显表现最佳。这说明 GWOQL 在 3 目标 16 组算例对比中相对其他比较算法显示出了很好的综合能力,反映了 GWOQL 良好的多样性和收敛性。

3.3 敏感性分析

3.3.1 样本量参数

在对不确定环境进行模拟时,所得样本会对结果产生影响。下面研究抽取样本数量的变化对结果的影响,本文设置原定样本量为 1 000,此处设置 300、500 以及 800 样本量的对照试验组,随机选取 3 个算例进行验证,表 5 为三组算例在选取不同样本量情况下所得的均值与方差结果。

通过表 7 的分析结果可以观察到,在样本量为 1 000 时,大多数实验组能够获得最佳的结果值。随着样本量的增加,用户完工时间、平台收益以及服务商利润均呈现出较好的结果。这是因为随着样本数量的增加,所得目标期望值越接近真实情

况下的解,从而优化效果也会变得更好。因此,选择样本量为 1 000 作为本文参数更符合用户、平台和服务商的期望。需要注意的是,在确定最佳样本量时,还需考虑成本和资源限制等因素,以平衡模型的精度和实际可行性。

3.3.2 波动范围

针对本文所涉及的任务加工时间,对于不同的现实情况,时间变化的程度也会有所不同,因此下面将要研究不同的波动范围对最终结果所产生的影响是否有所不同,本文设置不确定参数服务时间的波动范围为[0.8,1.2],另外增加[0.7,1.3],[0.6,1.4],[0.5,1.5]三种不同范围进行对比,并随机选取 4 个算例进行实验验证,所得结果如表 6 所示。

结果显示,在四种不同的波动范围情况下,多数实验组在[0.8,1.2]范围内的均值与目标值之间得到的结果具有最大的取值范围。这表明随着范围的变化,结果对最终优化目标的影响程度存在差异。在本文案例中,选择[0.8,1.2]作为实验范围能够更有效地实现优化目标。不同范围对结果的影响也可能存在差异,实际情况可能因环境、条件等因素而不同,因此在确定范围时需要充分的实证研究和分析,以确保模拟结果的准确性和可靠性。

表 5 不同样本量对比结果

Tab.5 Results of sample sizes

	标准差				均值			
	1 000	50	100	300	1 000	50	100	300
1	0.224 8	0.667 7	0.300 5	2.477 3	0.141 7	2.431 1	0.165 4	2.481 2
2	0.288 2	0.784 2	0.294 1	2.294 3	0.133 3	2.360 6	0.145 3	2.285 0
3	0.278 2	0.523 3	0.513 9	2.436 2	0.084 7	2.288 5	0.096 0	2.275 0
4	0.280 9	0.659 3	0.351 3	2.262 2	0.134 2	2.290 3	0.134 8	2.308 6
5	0.242 1	0.681 3	0.425 5	2.415 2	0.115 6	2.336 4	0.104 3	2.369 9
6	0.211 2	0.427 9	0.315 8	2.275 7	0.074 3	2.262 0	0.120 0	2.538 0
7	0.271 7	0.787 5	0.442 5	2.185 6	0.094 3	2.204 9	0.104 8	2.238 4
8	0.391 0	0.927 9	0.312 7	2.440 3	0.125 3	2.328 7	0.178 0	2.459 1
9	0.380 1	0.551 8	0.368 4	2.560 6	0.116 3	2.542 9	0.126 4	2.850 2
10	0.274 4	0.684 1	0.372 3	2.423 5	0.109 3	2.289 4	0.145 5	2.324 8
11	0.221 1	0.567 8	0.322 1	2.371 0	0.105 9	2.262 2	0.111 7	2.241 0
12	0.220 8	0.515 6	0.321 0	2.354 1	0.124 7	2.320 5	0.151 2	2.371 5
13	0.274 9	0.683 3	0.382 0	2.396 0	0.154 9	2.326 7	0.144 4	2.412 0
14	0.291 4	0.718 9	0.359 3	2.312 8	0.133 2	2.341 4	0.138 3	2.303 8
15	0.275 4	0.814 9	0.358 5	2.316 7	0.077 2	2.366 6	0.079 5	2.258 8
16	0.263 5	0.546 4	0.307 7	2.308 1	0.175 7	2.359 3	0.124 0	2.267 9

表 6 不同波动范围对比结果

Tab.6 Fluctuation range results of service time

	标准差				均值			
	0.8~1.2	0.5~1.5	0.6~1.4	0.7~1.3	0.8~1.2	0.5~1.5	0.6~1.4	0.7~1.3
1	0.224 8	0.667 7	0.360 9	0.959 7	0.424 9	0.815 5	0.323 9	0.653 4
2	0.288 2	0.784 2	0.306 8	0.770 4	0.430 8	0.925 4	0.524 6	0.892 3
3	0.278 2	0.523 3	0.440 4	0.794 6	0.551 7	1.139 4	0.430 8	1.083 3
4	0.280 9	0.659 3	0.405 1	0.787 7	0.455 2	0.745 9	0.461 7	0.952 7
5	0.242 1	0.681 3	0.358 9	0.924 2	0.350 9	0.887 2	0.351 0	0.848 3
6	0.211 2	0.427 9	0.245 6	0.974 5	0.228 2	0.624 6	0.303 5	1.090 0
7	0.271 7	0.787 5	0.464 4	1.185 1	0.412 9	0.596 1	0.454 7	1.158 7
8	0.391 0	0.927 9	0.298 5	0.832 2	0.455 2	0.796 9	0.570 0	0.933 3
9	0.380 1	0.551 8	0.373 9	0.552 2	0.346 6	0.606 9	0.367 3	0.672 9
10	0.274 4	0.684 1	0.415 0	0.876 0	0.500 0	0.990 1	0.500 5	0.778 4
11	0.221 1	0.567 8	0.280 0	0.683 5	0.289 5	0.626 7	0.272 1	0.836 3
12	0.220 8	0.515 6	0.310 0	0.502 2	0.400 4	0.675 1	0.288 7	0.548 1
13	0.274 9	0.683 3	0.443 3	1.006 7	0.401 9	0.964 4	0.516 7	0.761 0
14	0.291 4	0.718 9	0.315 2	0.763 1	0.313 8	0.647 5	0.429 4	1.176 8
15	0.275 4	0.814 9	0.418 6	0.929 5	0.256 8	0.673 5	0.376 8	1.059 8
16	0.263 5	0.546 4	0.292 8	0.767 9	0.335 8	0.745 2	0.238 7	0.623 5

4 结论

本文通过建立随机调度模型,在传统模型基础上考虑任务环境中的随机性与多方利益共同性,使模型与实际更加贴合,为云制造平台实现可持续发展提供管理启示。同时,将 GWO 算法与强化学习结合改进为算法 GWOQL, GWOQL 在与其他 6 种同类算法对比中显示出较强多样性与收敛

性,能够更加有效地权衡各个目标之间的关系,找到最优的平衡点,最大化同时满足用户、云平台以及服务商三方的需求。

参考文献:

[1] 周 济. 智能制造——“中国制造 2025”的主攻方向[J]. 中国机械工程, 2015, 26(17): 2273-2284.
[2] WANG J, XU C, ZHANG J, et al. Big data analytics for intelligent manufacturing systems: A review[J]. Journal

- of Manufacturing Systems, 2022, 62: 738-752.
- [3] PENG J, LEI R. The supply chain network on cloud manufacturing environment based on COIN model with Q-Learning algorithm [C]//2017 5th International Conference on Enterprise Systems (ES). IEEE, 2017: 52-57.
- [4] ZHANG J, DING G F, ZOU Y S, et al. Review of job shop scheduling research and its new perspectives under industry 4.0 [J]. Journal of Intelligent Manufacturing, 2019, 30(4): 1809-1830.
- [5] 陈康, 郑纬民. 云计算:系统实例与研究现状[J]. 软件学报, 2009, 20(5): 1337-1348.
- [6] QUY V K, HAU N V, ANH D V, et al. IoT-enabled smart agriculture: Architecture, applications, and challenges[J]. Applied Sciences-Basel, 2022, 12(7): 3396.
- [7] SHAKARAMI A, SHAKARAMI H, GHOBAEI-ARANI M, et al. Resource provisioning in edge/fog computing: A comprehensive and systematic review[J]. Journal of Systems Architecture, 2022, 122: 102362.
- [8] LU C, LIU Q, ZHANG B, et al. A pareto-based hybrid iterated greedy algorithm for energy-efficient scheduling of distributed hybrid flowshop[J]. Expert Systems with Applications, 2022, 204: 117555.
- [9] WANG J J, WANG L. A knowledge-based cooperative algorithm for energy-efficient scheduling of distributed flow-shop[J]. IEEE Transactions on Systems Man Cybernetics-Systems, 2020, 50(5): 1805-1819.
- [10] 张安, 杨咪, 毕文豪, 等. 基于多策略 GWO 算法的不确定环境下异构多无人机任务分配[J]. 航空学报, 2023, 44(8): 148-164.
- [11] GHAMGINZADEH A, NAJAFI A A, KHALILZADEH M. Multi-objective multi-skill resource-constrained project scheduling problem under time uncertainty[J]. International Journal of Fuzzy Systems, 2021, 23(2): 518-534.
- [12] 于冬梅, 高雷阜, 赵世杰. 考虑共享不确定因素的应急设施最大覆盖选址模型[J]. 运筹与管理, 2020, 29(12): 43-50.
- [13] 王天日, 张敏敏, 刘娟, 等. 考虑云制造服务协同的多用户任务调度优化[J]. 计算机集成制造系统, 2023, 29(9): 3006-3017.
- [14] DEB K, GUPTA H. Introducing robustness in multi-objective optimization [J]. Evolutionary Computation, 2006, 14(4): 463-494.
- [15] MCKAY M D, BECKMAN R J, CONOVER W J. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code [J]. Technometrics, 1979, 21(2): 239-245.
- [16] QIANG J, QI R B, QIAN F, et al. Multi-objective robust optimization based on NSGA-II and degree of robustness [C]//2010 8th World Congress on Intelligent Control and Automation. IEEE, 2010: 4859-4864.
- [17] 李稚, 周双牛. 面向绿色智能制造的高维多目标动态作业车间调度优化[J]. 运筹与管理, 2023, 32(1): 47-53.
- [18] 邝航宇, 金晶, 苏勇. 自适应遗传算法交叉变异算子的改进[J]. 计算机工程与应用, 2006(12): 93-96+99.
- [19] 李书全, 孙雪, 孙德辉, 等. 遗传算法中的交叉算子的述评[J]. 计算机工程与应用, 2012, 48(1): 36-39.

(责任编辑 唐彦彦)

《河北工程大学学报(自然科学版)》2024 年起由季刊变更为双月刊

为进一步缩短论文出版周期,更快推动科研成果的传播,满足广大专家学者学术交流需要,经主管主办单位同意,申请期刊刊期变更,获得河北省新闻出版局批复(冀新闻出版[2024]-060000001),《河北工程大学学报(自然科学版)》于 2024 年第 1 期起由季刊变更为双月刊,每逢双月的 25 号出版,特此公告。