

自适应 Jaya 算法求解多目标柔性车间绿色调度问题

王建华[†], 潘宇杰, 孙 瑞

(江苏大学 管理学院, 江苏 镇江 212013)

摘要: 针对多目标柔性作业车间绿色调度问题 (MO-FJGSP), 建立优化目标为最大完工时间、机器总负荷和能耗最小的多目标数学模型, 并设计了一种基于 Pareto 最优解的自适应多目标 Jaya 算法 (SAMO-Jaya) 对该问题进行优化求解. 该算法采用两级实数编码方式实现工序排序与机器分配的编码表示, 并设计一种转换机制实现将 Jaya 连续解空间映射至 FJSP 离散解空间; 然后设计了一种混沌序列与均匀分布相结合的混合策略以提高初始种群的质量与全局分散性; 此外, 在 Jaya 算法中嵌入了自适应调整种群规模的方法以提高算法求解速度. 通过 10 个单目标与 3 个多目标基准算例测试, 并与 7 个已有算法进行对比分析, 结果表明 SAMO-Jaya 算法能够对 MO-FJGSP 进行有效求解.

关键词: 多目标优化; 柔性作业车间绿色调度; 自适应多目标 Jaya 算法; Pareto 最优解; 混合策略

中图分类号: TP301

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.1773

开放科学 (资源服务) 标识码 (OSID):



Multi-objective flexible job shop green scheduling problem with self-adaptive Jaya algorithm

WANG Jian-hua[†], PAN Yu-jie, SUN Rui

(College of Management, Jiangsu University, Zhenjiang 212013, China)

Abstract: A mathematical model aiming at minimizing the makespan, total machine utilization and energy consumption is established according to the multi-objective flexible job shop green scheduling problem. A self-adaptive multi-objective Jaya algorithm (SAMO-Jaya) based on Pareto optimal is designed to optimize the model. And a two-level real number encoding is adopted to implement the coding scheme of processes' sequences and machines' assignment, then a transformation mechanism is designed to create a mapping between the continuous solution space of Jaya and the discrete solution space of FJSP. And then a hybrid strategy combining chaotic sequence and uniform distribution is raised to improve the quality and diversity of the initial populations. In addition, a self-adaptive population size adjusting method is embedded to improve the optimizing speed of the algorithm. By analyzing the solutions of 10 single-objective benchmarks and 3 multi-objective benchmarks solved by SAMO-Jaya and other 7 existing algorithms, the results show that SAMO-Jaya can solve the MO-FJGSP effectively.

Keywords: multi-objective optimization; flexible job shop green scheduling; self-adaptive multi-objective Jaya algorithm; Pareto optimal; mixed strategy

0 引言

生产力的快速发展, 导致能源的消耗速度日渐增长. 近些年来, 世界各国已经意识到了能源利用效率不高导致的浪费以及带来的环境污染问题. 绿色制造正是在这个环境下成为学术界与企业界所热议的话题^[1]. 而绿色车间调度作为绿色制造的关键环节之一, 也是引起了许多学者的兴趣与关注^[2].

多目标柔性作业车间调度问题 (Multi-objective Flexible Job-shop Scheduling Problem, MO-FJSP) 是对传统车间调度问题的扩展, 是较为经典的 NP-hard 问

题, 具有多指标、工艺路线柔性、加工机器不确定等特性, 更加符合实际的生产状态. 智能优化算法是求解 MO-FJSP 极为有效的一种方法, 并且已有部分学者运用这些算法取得了研究成果, 例如杂草算法、粒子群算法、萤火虫算法等. 黄霞等^[3] 建立以最大完工时间、总流经时间和总延迟时间为目标的置换流水车间调度问题模型, 并提出一种改进的混沌杂草算法进行求解. 吴锐等^[4] 提出了一种基于三维向量编码方式与局部搜索算子的改进人工蜂群算法用于解决分布式柔性车间调度问题. Dai 等^[5] 提出了一种改进

收稿日期: 2019-12-19; 修回日期: 2020-03-14.

基金项目: 国家自然科学基金项目 (71673118).

[†]通讯作者. E-mail: jiannywang@163.com.

遗传算法用于解决带有运输约束^[13]的多目标柔性作业车间调度问题。Amiri 等^[6]考虑资源分配和作业顺序不确定,建立多目标柔性作业车间模型,并用组合优化策略进行求解。Zhong 等^[7]以最小化最大模糊完成时间和最小化最大模糊机器负荷为调度目标建立多目标模糊柔性作业车间调度问题模型,并设计了一种改进人工蜂群算法对问题进行求解。韩玉艳等^[8]采用离散 NSGA-II 算法求解带有限缓冲区的多目标流水线调度问题。朱光宇等^[9]设计了一种基于直觉模糊集相似度的遗传算法用于解决考虑机器质量的多目标柔性作业车间调度模型。在上述研究中,研究者们大多只关注最大完工时间、拖期率等经济指标,少有关注能耗等绿色指标。近些年来,开始有学者研究绿色车间调度问题,艾子义等^[10]以总碳排放为优化目标,建立相应低碳柔性车间调度问题,并提出一种新型蛙跳算法进行求解。Gong 等^[11]提出了一种混合遗传算法求解具有工人和机器柔性的双柔性作业车间调度问题。吴秀丽等^[12]考虑机器多转速的因素,通过调节机器转速来设计综合考虑完工时间和能源消耗的低碳调度算法。Zheng 等^[13]提出了一种协同多目标果蝇优化算法用于求解资源约束无关并联机床绿色制造调度问题。

Jaya 算法是由 Rao 教授于 2016 年提出的一种基于群体的新兴智能优化算法^[14],该算法区别于其他优化算法,具有无参数运行、求解速度快、不易陷入局部最优解等优点。自 Jaya 算法被提出以来,已经被广泛应用于诸多领域,并取得了卓越的成果,例如图像识别、机械设计、机器学习等^[15-17]。由于该算法在许多领域表现出良好的性能,因此在近些年来被引入车间调度问题进行求解,但由于 Jaya 算法提出时间较晚,在车间调度方面的应用还较少,且仅有的研究也大多集中在流水车间调度问题领域^[18-19],少有应用于多目标柔性作业车间调度问题研究。综上所述,本文以最小化最大完工时间、最小化机器总负荷、最小化能耗为优化目标,构建了多目标柔性作业车间绿色调度问题模型 (Multi-objective Flexible Jobshop Green Scheduling Problem, MO-FJGSP)。并设计了一种基于 Pareto 最优解的自适应多目标 Jaya 算法 (Self-adaptive Multi-objective Jaya algorithm, SAMO-Jaya),SAMO-Jaya 采用两级实数编码方式表示工件信息,针对基本 Jaya 算法只能解决连续函数问题的局限性,设计了一种转换机制用于建立 Jaya 连续空间与 FJSP 离散空间的映射;为了提高初始种群的质量与全局分布性,设计了一种混沌序列和均匀分布相

结合的混合策略用于生成初始种群;接着计算种群的 Pareto 等级和拥挤度用于选择最优解与最劣解,根据位置更新公式、最优解和最劣解对其余候选解进行更新;最后设计了一种根据解集质量自适应调整种群规模的方法,提高了求解速度,避免算法过早收敛。通过基准算例仿真并与近期具有代表性的多个算法进行性能对比,验证了算法在求解 MO-FJGSP 上的有效性。

1 多目标柔性作业车间绿色调度问题

1.1 问题描述

MO-FJGSP 描述如下: n 个工件在 m 台机器上加工,工件 $i(i = 1, 2, \dots, n)$ 包含 q_i 道工序,其中不同工件的工序数量可以不等;每道工序可以在多台机器上选择加工,且不同机器的加工时间可以不等, M_{ij} 表示工件 i 的第 j 道工序的可选机器集, $M_{ij} \in \{1, 2, \dots, m\}$; S_{ij} 为工件 i 的第 j 道工序的开始加工时间; T_{ij}^k 为工件 i 的第 j 道工序在机器 k 上的加工时间; F_{ij} 为工件 i 的第 j 道工序的结束加工时间;优化目标包括最大完工时间 C_T , 机器总负荷 T_M 和能耗 E 。

在对问题建立模型之前,需要依据实际情况做一些合理的假设:

- (1) 每台机器在同一时刻只能加工一个工件;
- (2) 机器在运行过程中不会出现故障;
- (3) 工件一旦进入机器加工就不能被打断;
- (4) 每道工序仅能被一台机器加工一次;
- (5) 在 0 时刻,所有工件材料已经准备完毕;
- (6) 同一工件的不同工序存在紧前紧后关系。

为了方便读者对本文的理解,现对本文中出现的符号做如下表 1 定义。

表 1 符号定义

| 符号 | 定义 |
|------------|---|
| n | 工件数量 |
| m | 机器数量 |
| i, h | 工件号 |
| j, v | 工序号 |
| k | 机器号 |
| q_i | 工件 i 的工序数 |
| O_{ij} | 工件 i 的第 j 道工序 |
| M_{ij} | 工序 O_{ij} 的可选机器集 |
| X_{ij}^k | 工序 O_{ij} 是否在机器 k 上加工,为 0-1 变量 |
| Y_{hvi} | 工序 O_{hv} 与工序 O_{ij} 的加工顺序,为 0-1 变量 |
| S_{ij} | 工序 O_{ij} 的开始加工时间 |
| T_{ij}^k | 工序 O_{ij} 在机器 k 上的加工时间 |

| | |
|------------|------------------------------|
| F_{ij} | 工序 O_{ij} 的结束加工时间 |
| a_{ij}^k | 工序 O_{ij} 在机器 k 上的负载单位能耗 |
| b^k | 机器 k 的空载单位能耗 |
| M | 一个大正实数 |
| C_T | 最终完工时间 |
| T_M | 机器总负荷 |
| E | 能耗 |

1.2 模型建立

在实际生产中, 管理者往往要依据多个目标作出决策, 本文的调度模型权衡了经济指标和绿色指标两个方面, 考虑了最大完工时间 (C_T), 机器总负荷 (T_M) 和能耗 (E) 三个优化目标, 可以表示为 $\min(C_T, T_M, E)$, 其中最大完工时间为经济指标, 机器总负荷为了提高设备的使用寿命, 而能耗为绿色指标, 具体的计算公式如下.

$$\min C_T = \max(F_{ij}) = \max(S_{ij} + T_{ij}^k) \quad (1)$$

$$\min T_M = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^m T_{ij}^k \quad (2)$$

$$\min E = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^m [a_{ij}^k X_{ij}^k T_{ij}^k + b^k (\max(X_{ij}^k F_{ij}) - \min(X_{ij}^k S_{ij}) - X_{ij}^k T_{ij}^k)] \quad (3)$$

$$s.t. X_{hv}^k F_{hv} \leq X_{ij}^k S_{ij} + M(1 - Y_{hvij}), \forall i, j, k; \quad (4)$$

$$F_{ij} - S_{ij} = T_{ij}^k, \forall i, j, k; \quad (5)$$

$$\sum_{k=1}^m X_{ij}^k = 1, \forall i, j; \quad (6)$$

$$S_{i(j+1)} \geq F_{ij}, \forall i, j; \quad (7)$$

$$X_{ij}^k \in \{0, 1\}, \forall i, j, k; \quad (8)$$

$$Y_{hvij} \in \{0, 1\}, \forall h, v, i, j; \quad (9)$$

其中: 式 (1) 式 (3) 为目标函数, 分别表示最小化最大完工时间、最小化机器总负荷、最小化能耗; 式 (4) 表示每台机器在同一时刻只能加工一个工件; 式 (5) 表示工件一旦开始加工中途不能被打断; 式 (6) 表示每道工序仅能被一台机器加工一次; 式 (7) 表示同一工件的工序之间存在紧前紧后关系; 式 (8) 为 0-1 变量, 若工序 O_{ij} 在机器 k 上加工, 则 $X_{ij}^k = 1$, 否则 $X_{ij}^k = 0$; 式 (9) 为 0-1 变量, 若工序 O_{hv} 在工序 O_{ij} 之前加工, 则 $Y_{hvij} = 1$, 否则 $Y_{hvij} = 0$.

2 自适应多目标 Jaya 算法

基本 Jaya 算法仅应用于求解连续函数问题, 而 MO-FJGSP 为离散函数问题, 为求解该问题, 需要解

决连续函数与离散函数之间的转换, 因此本文设计了一种转换机制用于连续与离散问题的相互转换. 为了提升初始种群的质量, 设计了一种混沌序列与均匀分布相结合的混合策略. Jaya 算法的核心在于位置更新策略, 该策略需要通过获取每一代种群的最优解与最劣解实现, 而 MO-FJGSP 为多目标问题, 不易直接得到两个极端解, 因此本文通过计算所有个体非支配等级来进行分类, 再通过计算拥挤度从第一类与最后一类中选择出最优解与最劣解. 此外, 在算法中嵌入了一种自适应调整种群规模的方法用于加快算法的求解速度. 区别于其他基于群体的智能优化算法, SAMO-Jaya 不需要特定的参数控制种群的迭代, 例如遗传算法的变异率、交叉率等, 而合适的参数对于算法的寻优能力有着很大的影响, 因此运用 SAMO-Jaya 对问题进行求解时不容易陷入局部最优解, 且寻优能力更强.

2.1 编解码与转换机制

编码方式的合适与否直接决定算法运行的性能, 对最终的求解结果有着相当重要的影响. MO-FJGSP 可以看成两个子问题, 即工序排序以及机器分配. 针对该特点, 本文的编码方式采用两级实数编码, 第一级采用基于工序的编码方式, 第二级采用基于机器的编码方式, 分别对应工序排序以及机器分配. 为方便理解, 现有一个简单 3×4 的 MO-FJGSP 例子, 工序集为 $O_1 = \{O_{11}, O_{12}, O_{13}\}, O_2 = \{O_{21}, O_{22}\}, O_3 = \{O_{31}, O_{32}, O_{33}\}$, 机器集为 $m = \{m_1, m_2, m_3, m_4\}$. 图 1 为一个可行解的编码, 第一行为工序编号, 第二行为机器分配, 该编码表示工件 1 的第 1 道工序在机器 1 上加工, 工件 2 的第 1 道工序在机器 3 上加工, 以此类推. 其中第一行的数字表示工件号, 相同的工件号表示同一工件的前后工序, 第二行的数字表示机器号, 这种两级实数编码方式既能满足工艺路线的约束, 又能满足加工机器柔性的约束.

| | | | | | | | | |
|------|---|---|---|---|---|---|---|---|
| 工序编号 | 1 | 2 | 3 | 1 | 3 | 1 | 3 | 2 |
| 机器分配 | 1 | 3 | 2 | 3 | 2 | 1 | 4 | 4 |

图 1 编码示例

由图 1 可以看出, MO-FJGSP 中的解都为离散的实数, 而 Jaya 算法的个体位置更新都为连续值, 因此为了实现算法解集与调度解空间的相互映射, 本文在这里针对工序编号与机器分配设计了一种转换机制.

1) 工序编号的转换

将个体位置的连续值范围限定为 $[\alpha, \beta]$, 依据式

(10) 可以将工件编码转化为相应的位置向量. 图 1 中的工件编码一行可以转换为图 2 所示的位置元素, 范围为 [1,2].

$$Z(i) = \frac{(\beta - \alpha) * i}{n} + \alpha \quad (10)$$

其中: $Z(i)$ 为工件 i 对应的位置向量, n 为工件数.

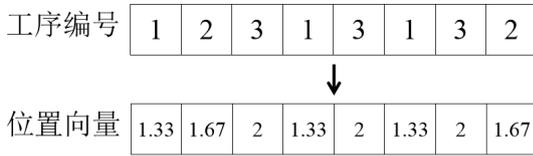


图 2 工序编号转换

2) 机器编号的转换

采用式 (11) 可以将可选机器集中的机器位置转换为个体位置向量, 连续值仍然限定为 $[\alpha, \beta]$, 假设上例中的可选机器集为 $M_1 = \{3, 2, 4\}, M_2 = \{1, 3\}, M_3 = \{4, 2, 4\}, M_1$ 表示工件 1 的三道工序可选机器集数量为 3, 2, 4, 以此类推. 图 1 中的机器分配可以转换为图 3 所示的位置元素.

$$Z(ij) = \frac{(\beta - \alpha) * pos(m_{ij})}{M_{ij}} + \alpha \quad (11)$$

其中: $Z(ij)$ 为工序 O_{ij} 所选机器 m_{ij} 的位置向量, $pos(m_{ij})$ 表示所选机器在工序可选机器集中的位置, M_{ij} 为工序 O_{ij} 的可选机器数量.

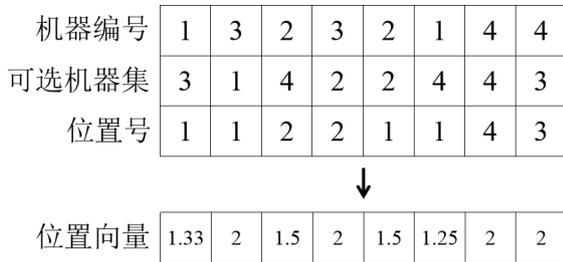


图 3 机器编号转换

3) 个体位置向调度方案的逆向转换

个体位置向调度方案的转换, 即将位置向量的连续值转换为调度方案的离散实数. 可以通过式 (10)、式 (11) 的逆运算式 (12)、式 (13) 实现, 位置向量转为工序编号可依据式 (12), 位置向量转为机器编号可依据式 (13), 然后根据求得的 $pos(m_{ij})$ 在可选机器集中确定机器编号, 结果均取整数.

$$i \approx \frac{[Z(i) - \alpha] * n}{\beta - \alpha} \quad (12)$$

$$pos(m_{ij}) \approx \begin{cases} \frac{[Z(ij) - \alpha] * M_{ij}}{\beta - \alpha} & pos(m_{ij}) \neq 0; \\ 1 & pos(m_{ij}) = 0. \end{cases} \quad (13)$$

需要注意的是, 在用式 (13) 进行运算时, $pos(m_{ij})$ 有可能会出现 0, 因此当 $pos(m_{ij})$ 为

0 时, 将 $pos(m_{ij})$ 默认为 1, 即选取 M_{ij} 中的第一个机器.

2.2 基于混合策略的种群初始化

初始种群的质量将对最终求得的结果产生很大的影响, 为了保证初始种群具有较高的质量, 同时保存种群的多样性, 本文设计了一种混沌序列与均匀分布相结合的混合策略产生初始种群. 混沌序列是一种确定系统中出现的无规则的运动, 但是值得注意的是混沌不是混乱无序, 而是有着精致内在结构且在一定范围内按其自身的“规律”不重复地遍历所有状态, 混沌行为的起源乃在于非线性系统对初值的敏感依赖性^[20]. 式 (14) 表明混沌序列的核心 logistic 方程, 其中 x_n 为一数列, 满足式 (14) 的递推关系.

$$x_{n+1} = vx_n(1 - x_n) \quad 2 < v < 4 \quad (14)$$

该操作主要分为两个阶段, 即工序排序与机器分配, 在工序排序阶段, 采用随机打乱的方法对所有工序的先后顺序进行排序; 在机器分配阶段, 每一道工序的机器都采用混沌序列的方法从可选机器集中产生, 在此过程中实时统计所有机器的被使用次数, 若某台机器的被使用次数到达平均值, 则该工序优先使用其他机器, 保证所有机器的负载均衡, 不断重复上述步骤, 直至生成完整的初始种群. 生成初始种群的具体步骤如下:

step 1: 定义调度方案的编码长度为 $\sum_{i=1}^n q_i$, 按顺序分别产生 q_i 个 i , 其中 $i = 1, 2, \dots, n$, 见图 4;

step 2: 对每个工序随机产生一个 $rand$ 值, 其中 $rand \in [0, 1]$, 根据 $rand$ 值的大小对所有工序进行升序排序, 可得到一组工序的加工序列, 见图 4;

step 3: 机器分配. 随机产生一组随机值 $Z(ij)$, 其中 $Z(ij) \in [\alpha, \beta]$, 根据式 (14) 的 logistic 方程可将 $Z(ij)$ 转为具有混沌序列特性的 $Z(ij)$, 再根据式 (13) 在可选机器集中进行选择;

step 4: 均匀分配. 统计所有机器的被使用次数, 若某一机器已达到被使用次数的平均值, 则该工序优先使用其他机器.

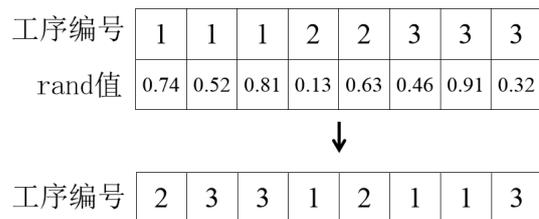


图 4 工序排序

2.3 计算个体非支配等级和拥挤度

对 MO-FJGSP 而言, 不能通过单一的目标来评价解的优良性, 为了选择出种群中的最优解与最劣解用于 SAMO-Jaya 的位置更新, 需要一种好的适应度评价策略来保证种群将表现优异的解遗传至下一代. 现有的研究中对于多目标问题的求解大多采用 Pareto 占优的方法, 求得非劣解集. 其中 Pareto 占优可以理解为对于 n 个目标分量 $f_i(x), i = 1, 2, \dots, n$, 任意给定两个决策变量 x_1, x_2 , 若满足以下两个条件, 则称 x_1 支配 x_2 .

- i) $\forall i \in 1, 2, \dots, n$, 都有 $f_i(x_1) \leq f_i(x_2)$ 成立;
- ii) $\exists i \in 1, 2, \dots, n$, 使得 $f_i(x_1) < f_i(x_2)$ 成立.

因此本文首先将采用 Pareto 占优计算个体的非支配等级, 在计算完个体等级之后, 还需考虑同一等级的 Pareto 解的优先级关系, 相关文献表明, 对于相同等级的多个解, 需要采用一定的策略来区分解之间的优先级关系, 目前较为常用的策略有聚类分析、信息熵和拥挤度等^[21]. 对于 SAMO-Jaya 算法而言, 对个体进行排序的主要目的是为了求得种群中最坏与最好的两个解用于其余个体位置的更新, 因此个体之间的拥挤度更加适用于本文算法. 拥挤度 n_d 的计算公式如式 (15) 所示.

$$n_d = \sum_{m=1}^M \frac{f_m(i+1) - f_m(i-1)}{f_m^{max} - f_m^{min}} \quad (15)$$

式中: M 为目标函数的个数, f_m^{max} 为个体目标函数值 f_m 的最大值, f_m^{min} 为个体目标函数值 f_m 的最小值, $f_m(i-1)$ 、 $f_m(i+1)$ 分别为该个体排序后的前后两位的目标函数值.

2.4 个体位置更新策略

在 SAMO-Jaya 算法的搜索过程中, 候选个体 $A_k(i)$ 由当前种群中最优解 $A_b(i)$ 和最劣解 $A_w(i)$ 引导进行个体位置的更新, 其中最优解与最劣解通过 2.3 小节的计算个体非支配等级和拥挤度来获得, 整个引导的过程是将候选个体引向最优解个体, 由于在更新过程中存在候选个体偏离最优解个体的情况, 因此在更新完个体后, 需要将新个体与旧个体的目标函数值进行比较, 选取表现优异的个体放入下一代. 个体位置更新公式如式 (16) 所示.

$$A_k(i+1) = A_k(i) + r_1(A_b(i) - |A_k(i)|) - r_2(A_w(i) - |A_k(i)|) \quad (16)$$

式中: r_1 、 r_2 为 $[0,1]$ 范围内随机产生的数字.

2.5 自适应调整种群规模

MO-FJGSP 作为一个实际生产中的问题, 所涉及的工件和机器数量都是非常庞大的, 因此种群的规模也十分巨大, 运用算法进行求解的速度相对较慢, 而在企业界, 对于算法的求解速度非常看重, 有时为了求解速度而放弃那些求解结果好但速度慢的算法. 因此为了提高 SAMO-Jaya 算法的求解速度, 本文嵌入了一种自适应调整种群规模的方法, 在算法时间复杂度不变的情况下, 通过有选择的缩小种群规模来达到提高求解速度的目的.

相关文献表明, 自 Jaya 算法被提出以来, 尚未有相关研究将其与自适应调整种群规模结合应用. 自适应调整种群规模的公式如式 (17) 所示, 其中 n_{new} 为下一代的种群规模, 结果取整数, n_{old} 为当前的种群规模, r 为 $[-0.3, 0.3]$ 之间的随机值, 将作为控制种群数量的参数. 在当代的种群质量好于上一代时, $r \in (0, 0.3]$, 种群规模将扩大, 当代的所有个体都进入下一代, 并且按照 2.3 小节所计算的 Pareto 等级和拥挤度对所有个体进行排名, 将当代种群前 $n_{new} - n_{old}$ 个体赋予剩余个体; 在当代的种群质量劣于上一代时, $r \in [-0.3, 0]$, 种群规模将缩小, 实行精英主义, 同样按照 2.3 小节计算所得种群个体排序, 选取前 $n_{old} - n_{new}$ 个体进入下一代; 在当代的种群质量与上一代相同时, $r = 0$, 种群规模不变.

$$n_{new} \approx n_{old} + r * n_{old} \quad (17)$$

其中, 本文对于评价种群质量的方法如下: 多目标问题中的单个目标之间不能直接进行对比, 不同代种群的所有个体目标值也无法直接进行对比, 因此文本评价种群质量的指标为 2.3 小节计算所得的非劣解的数量, 非劣解的数量是评价一个算法寻优性能的体现, 非劣解数量越多, 种群的质量越好, 当非劣解数量相同时, 视为两代种群质量相同.

2.6 算法流程

SAMO-Jaya 算法的具体实现步骤如下:

step 1: 种群初始化. 根据 2.2 节的方法进行种群初始化, 并设置参数迭代次数 $n = 0$ 与终止条件;

step 2: 寻找最优与最劣解. 根据 2.3 节的方法计算所有个体的非支配等级和拥挤度, 并进行排序选出其中的最优解与最劣解;

step 3: 更新种群. 根据 2.4 节的个体位置更新公式, 对所有候选解进行更新;

step 4: 判断新旧解. 新解是否好于旧解, 若是, 则将新解放入下一代; 否则将旧解放入下一代;

step 5: 判断终止条件. 终止条件是否满足, 若是,

输出最优解方案; 否则进行步骤 6;

step 6: 自适应调整种群规模. 根据 2.5 节的调整种群规模的方法对种群大小进行调整, 返回步骤 2, $n = n + 1$.

3 仿真测试与分析

本文采用 Matlab 2014a 进行编程, 并在 Windows7, Intel(R) Core(TM) i5-3337U, CPU1.8GHz, 内存 4GB, 64 位操作系统的计算机上运行. 算法的参数设置为: 种群大小为 100, 迭代次数为 200 代.

3.1 算法性能测试

为了验证 SAMO-Jaya 算法的可行性和有效性, 本文从单目标与多目标两个方面对本文算法进行测试. 在单目标方面, 本文将其应用于 Benchmark 算例中的 MK01-MK10 基准算例, 并用 SAMO-Jaya 算法分别运行 20 次, 取其中的最优解与具有相同优化理念的改进算法进行对比: Caldeira^[22] 提出的改进 Jaya 算法 (Improver Jaya Algorithm, IJaya)、Prasert 等^[23] 提出的改进微分进化算法 (Improved Differential Evolution, IDE) 和姜天华^[24-25] 提出的灰狼优化算法 (Grey Wolf Optimization, GWO) 以及混合灰狼优化算法 (Hybrid Grey Wolf Optimization, HGWO) 进行对比. 表 2 为各算法结果比较, 粗体表示相同算例中取得的最优解.

通过表 2 可以发现, 在 Benchmark 的十个算例中, 本文算法在算例 MK01 MK05 和算例 MK08 下都达到最优或优于其他 4 种算法, 能获得最优解, 只在 MK06/07/09/10 四个算例下未取得最优解, 但除了 MK10 的结果不理想外, 其余三个算例的结果都是接近于最优解. 此外, 数据显示 SAMO-Jaya 算法能获得最优解的个数为 6 个, 为五种算法中表现最好的算法. 图 5 为 MK01 算例下 Makespan=36 的调度方案. 其次, 与现有的 IJaya 算法相比, SAMO-Jaya 在 MK01-MK08 八个算例中, 有六个算例表现最好, 略优于 IJaya 的五个.

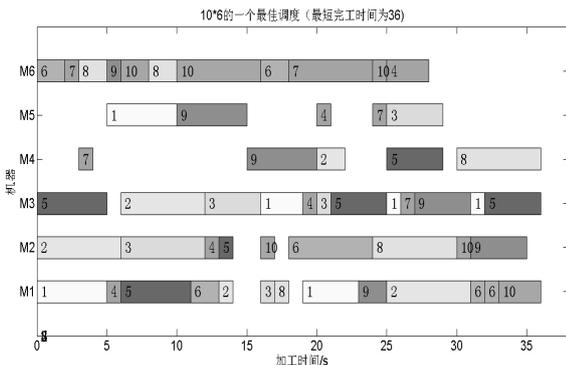


图 5 MK01(Makespan=36) 调度方案

表 2 Benchmark 算例对比

| 算例 | n×m | IJaya | IDE | GWO | HGWO | SAMO-Jaya |
|------|-------|------------|------------|------------|------------|------------|
| MK01 | 10×6 | 40 | 40 | 42 | 40 | 36 |
| MK02 | 10×6 | 28 | 28 | 31 | 29 | 28 |
| MK03 | 15×8 | 204 | 204 | 204 | 204 | 204 |
| MK04 | 15×8 | 60 | 71 | 68 | 65 | 59 |
| MK05 | 15×4 | 172 | 179 | 179 | 175 | 171 |
| MK06 | 10×15 | 57 | 73 | 68 | 79 | 75 |
| MK07 | 20×5 | 139 | 146 | 155 | 149 | 149 |
| MK08 | 20×10 | 523 | 528 | 523 | 523 | 523 |
| MK09 | 20×10 | - | 321 | - | 325 | 325 |
| MK10 | 20×15 | - | 235 | - | 253 | 272 |

为了探究各改进部分对算法的影响, 以 MK04 算例为例, 在此基础上分析改进部分对 Jaya 算法性能的影响, 图 6 为引入混合策略前后 Jaya 算法在求解 MK04 算例时的收敛曲线, 可以看出 SAMO-Jaya 算法在早期的收敛速度和收敛效果要明显优于 Jaya 算法, 且 SAMO-Jaya 算法在 125 代左右时已达到最优, 最优解为 59, 而 Jaya 算法直到 150 代左右才求得最优解 65, 由此可以看出本文设计的混合策略对提高种群质量的有效性. 为了验证在迭代过程嵌入自适应调整种群规模的方法之后对提高求解速度的有效性, 将 SAMO-Jaya 算法与 Jaya 算法同时求解 MK04 算例 20 次得到表 3 所示的性能比较结果, 其中 RT 为系统运行时间, VAR 为 20 次结果的方差, 可以看出 SAMO-Jaya 的求解速度明显快于 Jaya, 但在稳定性上面两者并没有差别. 因此, 在算法中嵌入自适应调整种群规模的方法后能有效提升求解速度, 但不会影响求解结果的稳定性.

表 3 Jaya 算法与 SAMO-Jaya 算法求解速度对比

| 指标 | RT/min | | VAR | |
|----|-----------|------|-----------|-------|
| | SAMO-Jaya | Jaya | SAMO-Jaya | Jaya |
| 结果 | 1.12 | 1.48 | 0.107 | 0.132 |

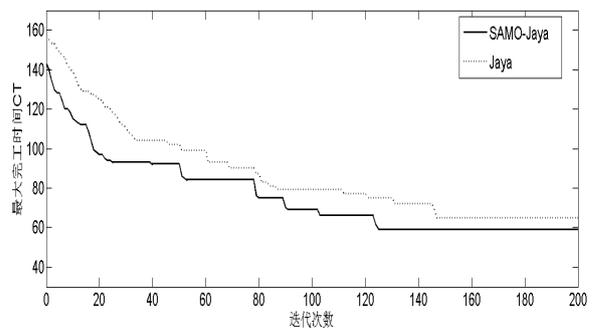


图 6 引入混合策略前后 Jaya 算法 MK04 收敛曲线

为了进一步验证本文算法的性能, 用 SAMO-Jaya 算法求解经典算例 Kacem 中的

8×8,10×10,15×10 三个基准算例, 并与 Alzahrani^[26] 提出的抢占式约束规则 (Pre-emptive Constraint Procedure, PCP)、曹磊^[27] 提出的变邻域杂草优化算法 (Variable Neighbourhood Invasive Weed Optimization, VNIWO)、Soto^[28] 提出的多目标分支界定算法 (Multi-objective Branch and Bound, MBB) 进行对比, 实验结果如表 4 所示. 其中 C_m 为最大完工时间, W_m 为瓶颈机器负载, W_t 为机器总负载.

从表 4 的实验结果可以看出, SAMO-Jaya 求解 8×8 问题所得的非劣解为 4 个, 仅次于 VNIWO, 多余 PCP 和 MBB, 将 SAMO-Jaya 与 VNIWO 的非劣解集放在一起进行 Pareto 占优操作后可以发现, VNIWO 还剩 [16,13,73] 一组解, 而 SAMO-Jaya 还剩 [15,11,77],[16,12,75],[15,12,76] 三组解, 因此 SAMO-Jaya 从解集质量来看要优于 VNIWO; 求解 10×10 问题所得非支配解为 4 个, 多余其余三个算法, 且 SAMO-Jaya 的性能指标更占优势; 求解 15×10 问题所得非支配解为 2 个, 少于 PCP 而多余 MBB. 因此, SAMO-Jaya 能够有效求解不同规模的多目标问题, 且能保证解的质量.

3.2 MO-FJGSP 求解与分析

求解本文问题模型所采用的算例为 MK08, 目标函数为最大完工时间 (C_T), 机器总负荷 (T_M) 和能耗 (E), 其中 C_T 和 T_M 均可由算例中的数据直接求得, E 的值无法直接求得, 因此添加各工件在机器上加工能耗以及各机器空载能耗的数据集, 由计算机

在 [0.5,2] 和 [0.1,0.3] 范围内随机产生, 14 台机器的单位加工能耗为 [1.3, 0.9, 1.7, 1.1, 1.2, 1.9, 0.7, 0.5, 2.0, 1.5, 1.8, 0.9, 1.1, 1.3], 14 台机器的单位空载能耗为 [0.2, 0.1, 0.1, 0.3, 0.2, 0.3, 0.2, 0.2, 0.2, 0.1, 0.2, 0.3, 0.1, 0.2]. 算法最终求得如表 5 所示具有 18 个非支配解的 Pareto 解集. 表 5 所示的 18 个解为本文问题模型提供了 18 种相对较优的调度方案, 在实际生产中, 生产计划员需要根据自身企业的需求, 通过层次分析法 (Analytic Hierarchy Process, AHP) 为不同的目标确定相应的权重, 从中选出最适用的调度方案, 假设 C_T 、 T_M 、 E 的权重分别为 0.5、0.3、0.2, 可以计算出上述 18 个非支配解得对应适应度值, 通过排序最终得到第 11 组为最优解. 最大完工时间为 534, 机器总负荷为 2567, 能耗为 647.3, 在迭代过程中各目标值的收敛曲线如图 7 所示, 第 11 组解的甘特图如图 8 所示.

表 5 最终代 Pareto 最优解集

| 序号 | C_T | T_M | E | 序号 | C_T | T_M | E |
|----|-------|-------|-------|----|-------|-------|-------|
| 1 | 536 | 2916 | 621.8 | 10 | 530 | 2815 | 697.5 |
| 2 | 547 | 2877 | 631.8 | 11 | 534 | 2567 | 647.3 |
| 3 | 527 | 2741 | 733.4 | 12 | 540 | 2713 | 535.3 |
| 4 | 543 | 2566 | 737.1 | 13 | 523 | 2884 | 712.5 |
| 5 | 533 | 2683 | 724.7 | 14 | 535 | 2654 | 637.9 |
| 6 | 550 | 2661 | 617.6 | 15 | 548 | 2742 | 621.2 |
| 7 | 530 | 2783 | 707.1 | 16 | 536 | 2739 | 636.5 |
| 8 | 534 | 2572 | 640.2 | 17 | 543 | 2563 | 740.2 |
| 9 | 539 | 2645 | 637.4 | 18 | 534 | 2765 | 718.6 |

表 4 Kacem 基准算例结果对比

| 算例 | 目标 | PCP | | | VNIWO | | | | MBB | | | SAMO-Jaya | | | | |
|-------|-------|-----|----|----|-------|----|----|----|-----|----|----|-----------|----|----|----|----|
| 8×8 | C_m | 16 | 15 | 16 | 15 | 16 | 17 | 17 | 18 | 15 | 17 | 14 | 15 | 16 | 15 | 17 |
| | W_m | 12 | 13 | 11 | 13 | 13 | 11 | 12 | 12 | 13 | 11 | 10 | 11 | 12 | 12 | 11 |
| | W_t | 78 | 76 | 82 | 75 | 73 | 78 | 76 | 75 | 73 | 75 | 78 | 77 | 75 | 76 | 75 |
| 10×10 | C_m | 8 | 9 | | 9 | 8 | 7 | | | 8 | 7 | | 7 | 8 | 7 | 7 |
| | W_m | 7 | 5 | | 7 | 5 | 6 | | | 6 | 7 | | 6 | 6 | 5 | 6 |
| | W_t | 44 | 43 | | 41 | 43 | 42 | | | 41 | 43 | | 42 | 41 | 43 | 42 |
| 15×10 | C_m | 13 | 11 | 11 | | 11 | 12 | | | 12 | | | 12 | 11 | | |
| | W_m | 10 | 11 | 10 | | 11 | 10 | | | 11 | | | 10 | 11 | | |
| | W_t | 93 | 95 | 98 | | 91 | 93 | | | 91 | | | 91 | 93 | | |

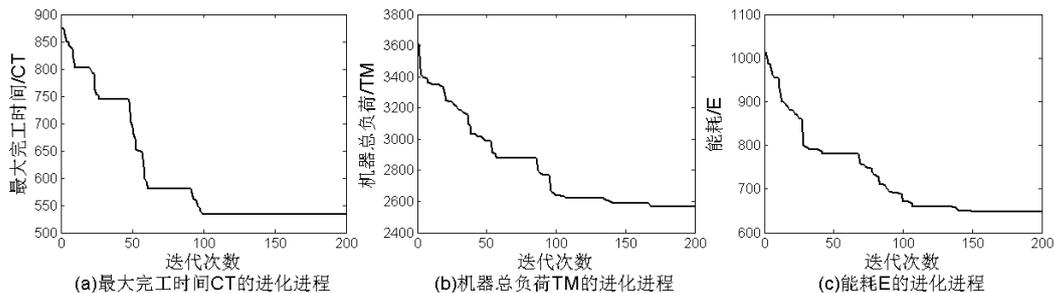


图 7 各目标函数最优值的进化过程

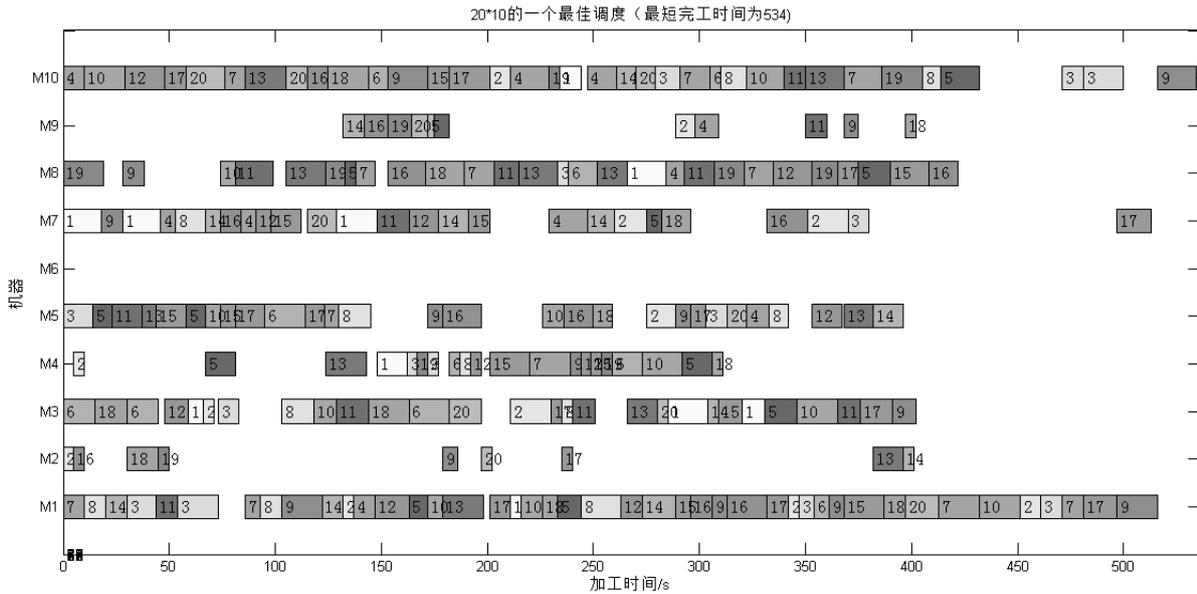


图8 最优解甘特图

4 结束语

针对能源消耗过大的现状,建立了以最大完工时间、机器总负荷、能耗为目标的多目标柔性作业车间绿色调度问题模型.并根据问题特点,对基本 Jaya 算法进行了一系列改进:1)设计了一种转换机制用于将问题从离散分布转换为连续分布,便于 Jaya 算法的求解;采用混沌序列与均匀分布相结合的混合策略产生初始种群,提高了初始种群的质量,同时降低了陷入局部最优解的概率;将自适应调整种群规模方法与 Jaya 算法相结合,加快了算法的求解速度.2)从单目标和多目标两个方面求解了 Benchmark 和 Kacem 的 13 个基准算例,对算法性能进行了验证,证明了算法的有效性和稳定性,同时求解了 MO-FJGSP 并对其进行了分析.最后,笔者将在本文研究基础上,进一步探究 Jaya 算法应用于动态重调度等更为复杂的问题.

参考文献 (References)

- [1] 中国工程院绿色制造发展战略研究. 推进绿色制造建设生态文明——中国绿色制造战略研究 [J]. 中国工程科学, 2017, 19(3): 53-60.
(The Research Group for Research on Green Manufacturing Strategy. Advancing Green Manufacturing for Ecological Civilization: Research on Green Manufacturing Strategies in China[J]. Chinese Engineering Science 2017, 19(3):53-60.)
- [2] 王凌, 王晶晶, 吴楚格. 绿色车间调度优化研究进展 [J]. 控制与决策, 2018, 33(3): 385-391.
(WANG L, Wang J J, WU C G. Advances in green shop scheduling and optimization[J]. Control and Decision, 2018, 33(3): 385-391.)
- [3] 黄霞, 叶春明, 曹磊. 多目标置换流水车间调度的混沌杂草优化算法 [J]. 系统工程理论与实践, 2017, 37(1): 253-262.
(HUANG X, YE C M, CAO L. Chaos invasive weed optimization algorithm for multi-objective permutation flow shop scheduling problem[J]. Systems Engineering - Theory & Practice, 2017, 37(1): 253-262.)
- [4] 吴锐, 郭顺生, 李益兵, 等. 改进人工蜂群算法求解分布式柔性作业车间调度问题 [J]. 控制与决策, 2019, 34(12): 2527-2535.
(WU R, GUO S S, LI L B, et al. Improved artificial bee colony algorithm for distributed and flexible job shop scheduling problem[J]. Control and Decision, 2019, 34(12): 2527-2535.)
- [5] DAI M, TANG D B, GIRET A S, et al. Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints[J]. Robotics and Computer-Integrated Manufacturing, 2019, 59: 143-157.
- [6] AMIRI F, BABAK S, ALI T. Multi-objective simulation optimization for uncertain resource assignment and job sequence in automated flexible job shop[J]. Applied Soft Computing Journal, 2019, 75: 190-202.
- [7] ZHONG Y G, YANG F, LIU F. Solving multi-objective fuzzy flexible job shop scheduling problem using MABC algorithm[J]. Journal of Intelligent & Fuzzy Systems, 2019, 36: 1455-1473.
- [8] 韩玉艳, 李俊青, 桑红燕, 等. 离散 NSGA-II 求解带有限缓冲区的多目标批量流水线调度问题 [J]. 聊城大学学报 (自然科学版), 2018, 31(1): 89-96.
(HAN Y Y, LI J Q, SANG H Y, et al. Discrete NSGA-II for Multi-objective Lot-streaming Flow Shop Scheduling Problem with Limited Buffers[J]. Journal of Liaocheng University(Nat.Sci.), 2018, 31(1): 89-96.)
- [9] 朱光宇, 徐文婕. 考虑能耗与质量的机床构件生产线

- 多目标柔性作业车间调度方法 [J]. 控制与决策, 2019, 34(02): 252-260.
(ZHU G Y, XU W J. Multi-objective flexible job shop scheduling method for machine tool component production line considering energy consumption and quality[J]. Control and Decision, 2019, 34(02): 252-260.)
- [10] 艾子义, 雷德明. 基于新型蛙跳算法的低碳柔性作业车间调度 [J]. 控制理论与应用, 2017, 34(10):104-111.
(AI Z Y, LEI D M. A novel shuffled frog leaping algorithm for low carbon flexible job shop scheduling[J]. Control Theory & Applications, 2017, 34(10): 104-111.)
- [11] GONG G L, DENG Q W, GONG X R, et al. A new double flexible job-shop scheduling problem integrating processing time, green production, and human factor indicators[J]. Journal of Cleaner Production, 2018(174): 560-576.
- [12] 吴秀丽, 孙阳君. 机器多转速的柔性作业车间绿色调度问题 [J]. 计算机集成制造系统, 2018, 24(04): 54-67.
(WU X L, SUN Y J. Flexible job shop green scheduling problem with multi-speed machine[J]. Computer Integrated Manufacturing Systems, 2018, 24(04): 54-67.)
- [13] ZHENG X L, WANG L. A Collaborative Multi-objective Fruit Fly Optimization Algorithm for the Resource Constrained Unrelated Parallel Machine Green Scheduling Problem[J]. IEEE Transactions on Systems, Man & Cybernetics, 2018, 48(5): 790-800.
- [14] RAO R V. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. International Journal of Industrial Engineering Computations, 2016(7): 19-34.
- [15] ZHANG Y D, YANG X J, CATTANI C, et al. Tea category identification using a novel fractional Fourier entropy and Jaya algorithm[J]. Entropy, 2016, 18(3): 77-93.
- [16] RAO R V, WAGHMARE G G. A new optimization algorithm for solving complex constrained design optimization problems[J]. Engineering Optimization, 2017, 49(1): 60-83.
- [17] SINHA R K, GHOSH S. Jaya based ANFIS for monitoring of two class motor imagery task[J]. IEEE Access, 2016(4): 9273-9282.
- [18] ASEEM M, DIVYA S. A TLBO and a Jaya heuristics for permutation flow shop scheduling to minimize the sum of inventory holding and batch delay costs[J]. Computers & Industrial Engineering, 2018, 124: 509-522.
- [19] BUDDALA R, MAHAPATRA S S. Improved teaching-learning-based and JAYA optimization algorithms for solving flexible flow shop scheduling problems[J]. Journal of Industrial Engineering International, 2018,14: 555-570.
- [20] 王宁, 蔚承建, 盛昭瀚. 基于嵌入混沌序列的遗传算法 [J]. 系统工程理论与实践, 1999, 19(11): 1-8.
(WANG N, WEI C J, SHENG Z H. Genetic Algorithm Combined with a Chaotic Sequence[J]. Systems Engineering - Theory & Practice, 1999, 19(11) :1-8.)
- [21] 肖海宁, 楼佩煌, 严伟国, 等. 柔性作业车间中机床与自动导引车在线调度方法 [J]. 农业机械学报, 2013, 44(4): 280-286.
(XIAO H N, LOU P H, YAN W G, et al. On-line Scheduling Method for Simultaneous Scheduling of Machines and Automated Guided Vehicles in Flexible Job Shop[J]. Transactions of the Chinese Society for Agricultural Machinery, 2013, 44(4): 280-286.)
- [22] Caldeira R H, Gnanavelbabu A. Solving the flexible job shop scheduling problem using an improved Jaya algorithm[J]. Computers and Industrial Engineering, 2019(137).<https://doi.org/10.1016/j.cie.2019.106064>.
- [23] Prasert S, Nuchara K, Preecha K. Improved Differential Evolution Algorithm for Flexible Job Shop Scheduling Problems[J]. Mathematical and Computational Applications, 2019, 24(3): 80-98.
- [24] 姜天华. 基于灰狼优化算法的低碳车间调度问题 [J]. 计算机集成制造系统, 2018, 24(10): 56-63.
(JIANG T H. Low-carbon workshop scheduling problem based on grey wolf optimization[J]. Computer Integrated Manufacturing Systems, 2018, 24(10): 56-63.)
- [25] 姜天华. 混合灰狼优化算法求解柔性作业车间调度问题 [J]. 控制与决策, 2018, 33(3): 503-508.
(JIANG T H. Flexible job shop scheduling problem with hybrid grey wolf optimization algorithm[J]. Control and Decision, 2018, 33(3): 503-508.)
- [26] Alzahrani J S. Multi-objective job shop scheduling using pre-emptive constraint procedure[J]. American Journal of Modeling and Optimization, 2019, 7(1): 8-13.
- [27] 曹磊, 叶春明, 黄霞. 基于员工学习行为的多目标柔性车间调度 [J]. 计算机集成制造系统, 2018, 24(08): 2023-2034.
(CAO L, YE C M, HUANG X. Multi-objective flexible job-shop scheduling based on learning effect[J]. Computer Integrated Manufacturing Systems, 2018, 24(08): 2023-2034.)
- [28] Soto C, Dorransoro B, Fraire H. Solving the multi-objective flexible job shop scheduling problem with a novel parallel branch and bound algorithm[J]. Swarm and Evolutionary Computation, 2020(53).
<https://doi.org/10.1016/j.swevo.2019.100632>.

作者简介

王建华(1977—), 男, 副教授, 博士, 从事工业工程、智能调度等研究, E-mail: jiannywang@163.com;

潘宇杰(1995—), 男, 硕士研究生, 从事算法优化、智能调度的研究, E-mail: wzupanyujie@163.com.

孙瑞(1995—), 男, 硕士研究生, 从事运筹学与智能算法的研究, E-mail: 772615743@qq.com.