



中国科技期刊卓越行动计划项目入选期刊

# 控制与决策

CONTROL AND DECISION



## 混合分解多目标进化算法求解绿色置换流水车间调度问题

罗聪, 龚文引

引用本文:

罗聪, 龚文引. 混合分解多目标进化算法求解绿色置换流水车间调度问题[J]. *控制与决策*, 2024, 39(8): 2737–2745.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2022.2145>

### 您可能感兴趣的其他文章

#### Articles you may be interested in

#### 区间数可重入混合流水车间调度与预维护协同优化

Collaborative optimization of interval number reentrant hybrid flow shop scheduling and preventive maintenance

*控制与决策*. 2021, 36(11): 2599–2608 <https://doi.org/10.13195/j.kzyjc.2020.0973>

#### 基于改进蛙跳算法的分布式两阶段混合流水车间调度

An improved shuffled frog leaping algorithm for the distributed two-stage hybrid flow shop scheduling

*控制与决策*. 2021, 36(1): 241–248 <https://doi.org/10.13195/j.kzyjc.2019.0472>

#### 超启发式交叉熵算法求解模糊分布式流水线绿色调度问题

Hyper-heuristic cross-entropy algorithm for green distributed permutation flow-shop scheduling problem with fuzzy processing time

*控制与决策*. 2021, 36(6): 1387–1396 <https://doi.org/10.13195/j.kzyjc.2019.1681>

#### 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 multi-objective flexible job shop scheduling problem

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

#### 自适应Jaya算法求解多目标柔性车间绿色调度问题

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

*控制与决策*. 2021, 36(7): 1714–1722 <https://doi.org/10.13195/j.kzyjc.2019.1773>

# 混合分解多目标进化算法求解绿色置换流水车间调度问题

罗 聪, 龚文引<sup>†</sup>

(中国地质大学 计算机学院, 武汉 430074)

**摘要:** 针对考虑能量消耗的绿色置换流水车间调度问题, 以最大完工时间和总能量消耗为优化目标, 提出一种混合分解多目标进化算法(HMOEA/D). 首先, 为了保持初始种群的多样性, 使用一种混合初始化策略产生高质量初始种群; 其次, 采用禁忌搜索策略作为局部搜索算子, 强化算法跳出局部最优能力; 最后, 提出节能策略, 以进一步优化总能量消耗目标. 通过对标准测试集进行仿真实验并与代表性算法进行比较, 验证所提出算法的优越性.

**关键词:** 置换流水车间调度; 绿色调度; 禁忌搜索策略; 节能策略; 分解多目标进化算法; 多目标优化

中图分类号: TP391

文献标志码: A

DOI: 10.13195/j.kzyjc.2022.2145

引用格式: 罗聪, 龚文引. 混合分解多目标进化算法求解绿色置换流水车间调度问题[J]. 控制与决策, 2024, 39(8): 2737-2745.

## A hybrid multi-objective evolutionary algorithm based on decomposition for green permutation flow-shop scheduling problem

LUO Cong, GONG Wen-yin<sup>†</sup>

(School of Computer Science, China University of Geosciences, Wuhan 430074, China)

**Abstract:** For the green permutation flow-shop scheduling problem (GPFSP) with energy consumption, a hybrid multi-objective evolutionary algorithm based on decomposition (HMOEA/D) is proposed, which aims at optimizing both of the makespan and total energy consumption. Firstly, in order to maintain the diversity of the initial population, a hybrid initialization strategy is used to initialize the population. Then, the tabu search strategy is used as local search operator to make the population jump out of local optima. Finally, an energy-saving strategy is proposed to further optimize the total energy consumption objective. Through the simulation of the standard benchmarks, and compared with the representative algorithm, the experimental results show that the proposed algorithm achieves better performance.

**Keywords:** permutation flow-shop scheduling; green scheduling; taboo search strategy; energy-saving strategy; multi-objective evolutionary algorithm based on decomposition; multi-objective optimization

## 0 引言

随着全球经济的高速发展, 调度在生产系统的决策和智能制造的资源分配上起着举足轻重的作用, 高效的调度策略是提高企业效率的关键<sup>[1-3]</sup>. 置换流水车间调度问题 (permutation flow-shop scheduling problem, PFSP) 作为调度领域中的经典问题而被广泛研究, 具有很重要的应用价值<sup>[4-5]</sup>. PFSP作为经典车间调度问题 (job shop scheduling problem, JSP) 的扩展, 较比于传统的车间调度问题更为常见, 已经成为工厂中典型的调度问题, 不仅每个工件需要在所有机器上进行加工, 而且工件在每台机器上的加工顺序相

同, 已被证明为NP-hard问题<sup>[6-7]</sup>.

PFSP具有良好的扩展性, 可与各种实际问题相结合, 由此产生许多复杂多样的调度问题. 因此, 在实际调度问题中, 会遇到许多的资源约束, 例如: 加工过程中存在阻塞<sup>[8]</sup>、缓冲区有限<sup>[9]</sup>、无空闲调度<sup>[10]</sup>等. 此外, 还存在许多不同模式的调度问题, 例如: 分布式流水车间调度问题<sup>[11-12]</sup>、流水车间组调度问题<sup>[13-14]</sup>等.

目前, PFSP在现代制造业和交通运输系统中起着至关重要的作用, 已有许多国内外学者致力于PFSP的研究. 近年来, 群智能优化算法和智能技术

收稿日期: 2022-12-12; 录用日期: 2022-06-25.

基金项目: 国家自然科学基金项目(62076225); 湖北省自然科学基金项目(2019CFA081).

责任编辑: 唐加福.

<sup>†</sup>通讯作者. E-mail: wygong@cug.edu.cn.

\*本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

飞速发展,许多新的启发式方法被广泛运用于求解PFSP中。Tian等<sup>[15]</sup>提出了新型布谷鸟算法求解PFSP,该算法提出一种新的移动策略,以增加解的多样性,从而避免陷入局部最优;Khurshid等<sup>[16]</sup>提出了混合进化算法,结合模拟退火算法高效求解PFSP;Ouchiekh等<sup>[17]</sup>提出了一种正余弦优化算法结合设计的鹰策略,通过实验验证了该算法求解PFSP的高效性;Ni等<sup>[18]</sup>设计的和声搜索算法结合模拟退火机制同样能很好地求解鲁棒PFSP。不仅如此,一些深度学习方法也被运用到求解PFSP中。Pan等<sup>[19]</sup>设计了一种新的深度神经网络求解PFSP,该方法使用演员批评法训练所设计的深度神经网络,并设计一种新的改进策略提炼最优解;Pavelski等<sup>[20]</sup>设计了一种抽样局部最优网络,采用4种采样策略有效求解PFSP。

随着全球温室效应的加剧和极端天气的频繁发生,绿色制造引起了各界人士的广泛关注<sup>[21-23]</sup>。由于制造业是最大的能源消耗产业,能量消耗往往是不可忽视的一环,节能调度在绿色制造系统中发挥了关键作用,由此产生了考虑总能量消耗目标的绿色置换流水车间调度问题(green permutation flow shop-scheduling problem, GPFSP)。为了达到节能减排的目的,在GPFSP中设计一种高效的节能策略是非常有意义的。

基于分解的多目标进化算法(multi-objective evolutionary algorithm based on decomposition, MOEA/D)是一种经典的多目标优化算法,该算法原理易懂、可扩展性高、参数设定少,具有较强的优化性能,受到了调度领域中诸多学者的关注<sup>[24]</sup>。近年来,MOEA/D被用于求解多目标置换流水问题<sup>[25-26]</sup>,它可以多目标问题分解为一系列的子问题进行优化,并采用切比雪夫聚合函数引导种群收敛。但是,MOEA/D对初始解比较敏感,初始解质量的好坏将直接影响算法的性能。同时,MOEA/D在处理复杂的调度问题时往往会陷入局部最优,其性能将会大打折扣。因此,根据GPFSP的特性,结合MOEA/D的优势并在其基础上设计针对特定问题的搜索算子以便有效求解GPFSP。

针对上述问题,本文提出一种混合基于分解的多目标进化算法(hybrid multi-objective evolutionary algorithm, HMOEA/D)求解绿色置换流水车间调度问题,同时优化最大完工时间和总能量消耗两个目标。该混合算法的主要特点是:1)融合3种初始化原则提出一种混合初始化策略来初始化种群,在保证种群多样性的前提下,生成高质量的初始种群;2)采用

禁忌搜索<sup>[27]</sup>策略作为局部搜索算子,设计特定的邻域结构与藐视准则帮助种群跳出局部最优,在避免重复搜索解空间的同时充分探索解空间,使种群获得更优的非支配解集;3)提出一种高效的节能策略,通过延迟工件的开始加工时间,在确保不增加最大完工时间的基础上,减小机器的空闲时间,进而优化总能量消耗目标。基于以上描述,HMOEA/D在MOEA/D的框架上,结合混合初始化策略提升算法的多样性。同时,设计特定的局部搜索算子和高效的节能策略提升算法的收敛性。相比于MOEA/D,HMOEA/D在保留MOEA/D较强优化性能的同时,可提高跳出局部最优的能力,进而充分探索解空间中的更优解。

## 1 绿色置换流水车间调度问题

### 1.1 问题描述

GPFSP是一类考虑生产成本和生产效率的资源分配问题,可以描述为: $n$ 个工件在 $m$ 台机器上进行加工,每个工件在每台机器上的加工顺序相同,每台机器上的各个工件的加工顺序也相同。同时,加工机器的类型相同,因而其加工功率和空闲功率也相同<sup>[28]</sup>。每个工件每次只能被一台机器加工,每台机器每次也只能加工一个工件。一旦工件开始加工就不允许被中断,各工件在各机器上的加工时间已知,期望获得一些最大完工时间最小、总能量消耗最少的最优调度序列。

### 1.2 数学模型

GPFSP的参数及决策变量定义如下。

参数:

$J$ : 工件集合,  $J = \{J_1, J_2, \dots, J_n\}$ ,  $n$ 为工件数;

$M$ : 机器集合,  $M = \{M_1, M_2, \dots, M_m\}$ ,  $m$ 为机器数;

$p_{i,j}$ : 工件 $J_j$ 在机器 $M_i$ 上的加工时间;

$S_{i,j}$ : 工件 $J_j$ 在机器 $M_i$ 上的开始加工时间;

$C_{i,j}$ : 工件 $J_j$ 在机器 $M_i$ 上的完工时间;

$C_{\max}$ : 最大完工时间;

$P^{\text{run}}$ : 机器加工时的功率;

$P^{\text{idle}}$ : 机器空闲时的功率;

$EC_r$ : 机器加工时的能量消耗;

$EC_s$ : 机器空闲时的能量消耗;

TEC: 总能量消耗;

$L$ : 充分大正数。

决策变量:

$x_{j,j'}$ : 0-1决策变量,若工件 $J_j$ 是工件 $J_{j'}$ 的紧前工件,则 $x_{j,j'} = 1$ ;否则 $x_{j,j'} = 0$ 。

总能量消耗主要由两部分组成:加工时的能量消耗和空闲时的能量消耗,其定义如下:

$$TEC = EC_r + EC_s. \tag{1}$$

加工时的能量消耗定义如下:

$$EC_r = \sum_{i=1}^m \sum_{j=1}^n p_{i,j} \cdot P^{run}. \tag{2}$$

空闲时的能量消耗定义如下:

$$\begin{cases} EC_s = \sum_{i=1}^m P^{idle} \sum_{j,j'=1}^n (S_{i,j'} - C_{i,j}); \\ x_{j,j'} = 1, \forall j, j' = 1, 2, \dots, n. \end{cases} \tag{3}$$

GPFSP的数学模型如下:

$$\begin{aligned} \min F_1 = C_{max} = \max C_{i,j}, \\ \forall i = 1, 2, \dots, n, \forall j = 1, 2, \dots, m; \end{aligned} \tag{4}$$

$$\min F_2 = TEC. \tag{5}$$

$$x_{j,j'} + x_{j',j} \leq 1, \forall j, j' = 1, 2, \dots, n; \tag{6}$$

$$\begin{aligned} C_{i,j'} - p_{i,j'} - C_{i,j} + L \cdot (1 - x_{j,j'}) \geq 0, \\ \forall i = 1, 2, \dots, m, j, j' = 1, 2, \dots, n; \end{aligned} \tag{7}$$

$$\begin{aligned} C_{i+1,j} - p_{i+1,j} \geq C_{i,j}, \\ \forall i = 1, 2, \dots, m - 1, j = 1, 2, \dots, n; \end{aligned} \tag{8}$$

$$x_{j,j'} \in \{0, 1\}, \forall j, j' = 1, 2, \dots, n. \tag{9}$$

其中:式(4)和(5)表示优化目标函数为最大完工时间和总能量消耗,式(6)和(7)表示一台机器在同一时刻只能加工一个工件,式(8)确保工件在前一台机器上加工完才能在下一台机器上加工,式(9)定义了变量的取值范围.

## 2 混合MOEA/D算法

### 2.1 编码与解码

本文的编码采用基于置换的编码.种群中每一个个体表示问题的一个解,每个个体由 $n$ 个无重复排列的工件组成,工件号的排列顺序表示在每台机器上的加工顺序.如图1所示,以10个工件、5台机器的问题为例,个体(10, 3, 2, 4, 5, 1, 7, 9, 8, 6)表示在第1台机器上工件的加工顺序为:10, 3, 2, 4, 5, 1, 7, 9, 8, 6,同样,在第2台机器上也是如此,以此类推,直到所有工件被第5台机器加工完.

工件序列 

10	3	2	4	5	1	7	9	8	6
----	---	---	---	---	---	---	---	---	---

图1 编码与解码

### 2.2 种群初始化

在求解多目标问题时,初始种群的质量是影响种群收敛速度和最终结果的关键,为了获得高质量的

种群,往往采用启发式算法初始化种群.启发式算法利用问题的特定知识和经验生成初始种群,可以在短时间内获得次优解,虽然启发式算法无法获得加工工件的最佳处理序列,但可以在一定程度上保证加工工件处理序列的局部最优性.本文结合3种初始化策略:NAWAZ-ENSCORE-HAM(NEH)原则<sup>[29]</sup>、总能量消耗最小原则、随机初始化原则,提出一种混合初始化策略.

1) NAWAZ-ENSCORE-HAM原则:将所有工件按照最长处理时间降序排序,从当前序列中取出前两个工件并将其排列,可以得到两个部分调度序列,评价这两个部分调度序列并保留最大完工时间最小的序列作为当前调度序列.取出下一个工件,将其插入到该调度序列中所有可能的位置,得到一些部分调度序列,评价各个部分调度序列并保留最大完工时间最小的序列作为当前调度序列.以此类推,直到遍历完所有工件,得到最大完工时间最小的调度序列.

2) 总能量消耗最小原则:将所有工件按照最长处理时间降序排序,从当前序列中取出前两个工件,依次将剩余工件插入到最合适的位置,得到总能量消耗最小的序列.

3) 随机初始化原则:随机生成无重复长度为 $n$ 的序列,保证种群的多样性.

采用NEH原则和总能量消耗最小原则生成两个较优个体,引领种群向更好的方向探索解空间,其余个体采用随机初始化原则随机生成.

### 2.3 遗传操作

遗传操作可以交换个体之间的信息,在一定程度上可以达到跳出局部最优的作用.由于工件的数量是确定的,不能像实数域中的交叉方式一样直接交换两个工件序列片段,染色体的结构必须满足以下两个约束:1)工件序列的每个基因值表示工件号,取值范围必须是 $1, 2, \dots, N$ 的范围内的离散值.2)工件序列中的每个基因值不能重复.在进行遗传操作时需要保证产生的新解仍然满足以上两个约束.针对于传统遗传操作会违背以上约束的问题,文献[30]提出了一种的遗传操作算子(IPOX),在不违背两个约束的前提下交换不同染色体中的信息.

假设有10个工件 $J_1, J_2, \dots, J_{10} \in J$ ,每个工件号 $J_i$ 只在染色体中出现一次,遗传操作的具体步骤如下.

step 1: 随机将工件集合 $J$ 划分成两个子集 $A, B$ .如图2所示,子集 $A = \{J_1, J_3, J_4, J_6, J_8, J_9\}$ ,子集 $B$

$= \{J_2, J_5, J_7, J_{10}\}$ .

step 2: 选取两个原解1和原解2,将原解1中属于集合A的工件号拷贝到新解1中,将原解2中属于集合B的工件号拷贝到新解2中.

step 3: 将原解1中属于集合A的工件号从左到右依次填充到新解2中,同样地将原解2中属于集合B的工件号从左到右依次填充到新解1中.

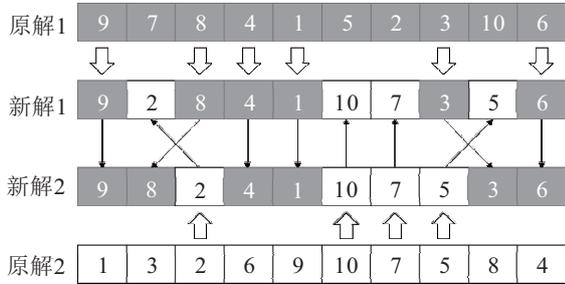


图2 遗传操作

2.4 禁忌搜索策略

禁忌搜索策略是对局部搜索的一种扩展,通过禁忌表来避免重复搜索解空间,改进局部搜索容易陷入局部最优的不足.由于禁忌搜索策略具有灵活的邻域结构和藐视准则,在搜索过程中也可以接受劣势解,在提高搜索的过程中跳出局部最优的能力,从而得到更好的全局最优解.因此,禁忌搜索策略是一种局部搜索能力很强的全局寻优算法.

针对GPFSP存在多个分布不规则的局部最优解的特点,在执行完遗传操作后,采用禁忌搜索策略跳出局部最优,从而进一步提升算法的收敛性.同时,将禁忌搜索策略嵌入HMOEA/D中作为局部搜索算子,并对种群中的每个个体都执行局部搜索,通过藐视准则有选择地接受进化过程中产生的新解,进而在一定程度上避免搜索过程中陷入局部最优的问题.禁忌搜索策略设计如下.

邻域结构:在搜索过程中,使用互换的邻域搜索结构,随机交换工件序列中的两个工件.

藐视准则:在禁忌搜索策略中,可能会存在优于当前解的候选禁忌解,或者出现候选解全部被禁忌的情况,此时采用以下藐视准则,实现更高效的优化性能.

- 1) 若候选解优于当前解,则解禁该候选解并替代当前解,然后将其加入禁忌表,更新禁忌表.
- 2) 若所有候选解不存在优于当前解的情况,则从候选解中选出最优解,然后将其作为禁忌对象并加入种群中进行后续搜索,更新禁忌表.
- 3) 若所有候选解都被禁忌,则解禁候选解中的最优解,更新禁忌表.

2.5 节能策略

节能策略可以优化总能量消耗目标,降低总能量消耗值.本文提出一种节能策略,旨在不增加最大完工时间的前提下减小总能量消耗.

影响总能量消耗值的因素有两个:加工时间和空闲时间,而加工时间是固定不变的,因此想要减少总能量消耗值就只能通过延迟工件的开始加工时间来达到.然而,并不是只要存在空闲时间就可以延迟工件的开始加工时间,延迟工件的开始加工时间必须满足该工件在该机器上的完工时间小于该工件在下一台机器上的开始时间.从最后一台加工机器逆向前遍历机器上的所有加工工件直至第1台加工机器,对于同一机器上的加工工件,按照开始加工时间逆序依次遍历,如果该工件之后存在空闲时间并且满足调整该工件的开始加工时间后,其完工时间仍小于该工件在下一台机器上的开始加工时间,则延迟该工件的开始加工时间.

如图3所示:首先按照开始加工时间逆序遍历M<sub>3</sub>上的所有工件,得到工件5满足延迟开始加工时间的条件;然后按照开始加工时间逆序遍历M<sub>2</sub>上的所有工件,工件5之后存在空闲时间,并且满足调整工件5的开始加工时间后,其完工时间仍然小于在M<sub>3</sub>上的开始加工时间,则延迟工件5的开始加工时间;最后再按照开始加工时间逆序遍历M<sub>1</sub>上的所有工件.改善后的工件序列比原始工件序列具有更短的总空闲时间,总能量消耗值也相应地减小.因此,所提出的节能策略可以确保在不增加最大完工时间的前提下减小总能量消耗.

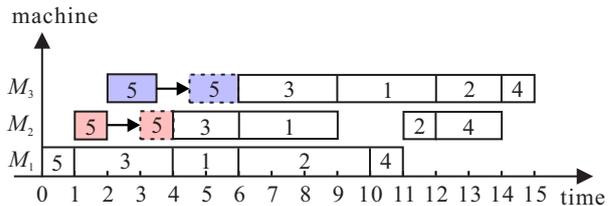


图3 节能策略

2.6 HMOEA/D算法描述

本节将MOEA/D算法与前文提到的种群初始化策略、遗传操作、禁忌搜索、新的节能策略相结合,提出混合基于分解的多目标进化算法(HMOEA/D).算法的基本流程如下.

step 1: 初始化N个均匀分布的参考向量,计算向量间的欧几里得距离,得到每个向量的T个邻域向量,将邻域信息存储到邻域矩阵中.

step 2: 使用3种初始化策略初始化N个种群个

体,计算适应值并更新全局参考点.

step 3: 对种群中的所有个体执行遗传操作,产生新解  $X_{new1}$  和  $X_{new2}$  并计算他们的适应值. 同时,为了维持种群数量,选取适应值较优的个体更新种群个体.

step 4: 根据设计的邻域结构和藐视准则执行禁忌搜索策略,搜索解空间的更优解.

step 5: 使用节能策略,在不增加最大完工时间的前提下,进一步优化总能量消耗目标值.

step 6: 根据邻域矩阵找出当前解  $X$  的  $T$  个邻域解,计算当前解与邻域解的切比雪夫聚合函数,若当前解优于邻域解则替换邻域解,否则邻域解不变.

step 7: 更新种群的非支配解集 PF. 若满足终止条件,则算法结束;否则,返回到 step 3.

## 2.7 时间复杂度分析

假设优化目标函数个数为  $M$ , 种群规模为  $N$ , 禁忌搜索迭代次数为  $T$ . 本文算法在迭代过程中主要由 3 部分组成: 第 1 部分为计算目标函数值,其时间复杂度为  $O(MN)$ ; 第 2 部分为禁忌搜索,其时间复杂度为  $O(NT)$ ; 第 3 部分为更新非支配解集,一个解放入非支配解集中最多需要与  $N$  个解进行比较,其时间复杂度为  $O(N^2)$ . 因此,本文算法的时间复杂度为  $O(N(M + T + N))$ .

## 3 实验结果与分析

### 3.1 实验设计

为了证明所提出的 HMOEA/D 解决 GPFSP 的有效性,本文算法和所有对比实验算法都采用 Matlab 编码,在 Intel Core i5-7500 CPU@3.40 GHz 3.41 GHz 和 8 GB RAM 的 Windows 10 操作系统的电脑上运行. 为了确保公平性,所有算法独立运行 20 次,通过 20 次运行的平均结果验证 HMOEA/D 的有效性.

对比算法主要分为 3 类: 1) 基于参考线的多目标算法,包括 MOEA/D<sup>[31]</sup> 和 AdaW(adaptation evolutionary algorithm)<sup>[32]</sup>; 2) 基于支配关系的多目标算法,包括 NSGA-II(fast and elitist multiobjective genetic algorithm)<sup>[33]</sup>; 3) 基于种群的算法,包括 AMEA (novel evolutionary algorithm with adaptation mechanism)<sup>[34]</sup>.

为了评估 HMOEA/D 解决 GPFSP 的有效性,选用超体积度量 (hypervolum, HV)<sup>[35]</sup> 评价指标,计算公式如下:

$$HV(P, R) = \bigcup_{p \in P} v(p, R). \quad (10)$$

其中:  $P$  表示算法求得的非支配解集;  $R$  表示参考点,在归一化非支配解集  $P$  后设置为  $R = (1.01, 1.01)$ ;  $v$  表示非支配解集  $P$  与参考点  $R$  之间形成的超立方体体积; HV 表示在目标空间中算法获得的非支配解集  $P$  与参考点  $R$  包围的超立方体体积的并集, HV 值越大,算法的综合性能越好.

### 3.2 实验数据集

实验测试数据选自文献[36],该数据集都是置换流水车间调度的标准测试集,其中机器的加工功率为 5,空闲功率为 2,分为两种规模的测试集.

一种是小规模测试集,共有 9 组数据,每组数据有 10 个实例. 最小规模的实例工件数是 20,机器数是 5;最大规模的实例工件数是 100,机器数是 20.

另外一种是大规模测试集,共有 3 组数据,每组数据有 10 个实例. 最小规模的实例工件数是 200,机器数是 10;最大规模的实例工件数是 500,机器数是 20.

### 3.3 实验参数

HMOEA/D 有 4 个重要参数,分别是: 种群大小、邻域个数、交叉概率和迭代次数. 为了确定最佳的参数组合,本文采用田口正交实验<sup>[37]</sup>. 每个参数的取值如下: 种群大小 = {40, 60, 80, 100}; 邻域个数 = {5, 10, 15, 20}; 交叉概率 = {0.7, 0.8, 0.9, 1.0}; 迭代次数 = {125, 150, 175, 200}. 图 4 展示了 HV 值主效应, HV 值越大代表参数的取值越好.

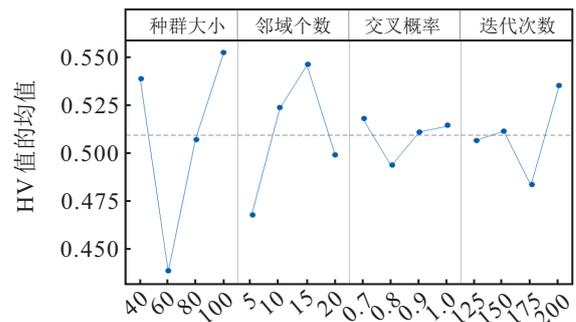


图 4 HV 值主效应

基于以上分析, HMOEA/D 的参数设置如下: 种群大小为 100, 邻域个数为 15, 交叉概率为 0.7, 迭代次数为 200, 禁忌搜索迭代次数为 50. 为了确保公平性, 对比算法的参数设置均参考原文献, 其中: AdaW 的种群规模大小为 100, 邻域个数为 10, 交叉概率为 1.0, 迭代次数为 200; AMEA 的种群规模为 50, 迭代次数为 200; MOEA/D 的种群规模为 100, 邻域个数为 15, 交叉概率为 0.7, 迭代次数为 200; NSGA-II 的种群规模为 100, 交叉概率为 0.7, 迭代次数为 200.

### 3.4 初始化策略的有效性

为了验证种群初始化策略的有效性,本文编码两种算法,一种是结合简单的随机初始化策略的HMOEA/D+Random,另外一种结合3种初始化策略的HMOEA/D+HI,其他所有参数设置相同,在所有测试数据集上独立运行20次并统计HV的平均值和最好值,结果如表1所示。

表1 HMOEA/D+Random和HMOEA/D+HI对比结果

实例	HMOEA/D+Random		HMOEA/D+HI	
	平均值	最好值	平均值	最好值
(20, 5)	0.704 815	0.838 694	<b>0.720 556</b>	<b>0.842 425</b>
(20, 10)	0.826 061	0.846 072	<b>0.830 772</b>	<b>0.852 812</b>
(20, 20)	0.799 502	0.848 980	<b>0.801 956</b>	<b>0.856 796</b>
(50, 5)	0.642 479	0.686 931	<b>0.653 746</b>	<b>0.704 331</b>
(50, 10)	0.808 178	0.875 856	<b>0.811 204</b>	<b>0.881 883</b>
(50, 20)	0.756 829	0.814 048	<b>0.758 611</b>	<b>0.826 213</b>
(100, 5)	0.671 882	0.738 235	<b>0.690 385</b>	<b>0.778 943</b>
(100, 10)	0.718 193	0.791 900	<b>0.720 720</b>	<b>0.793 542</b>
(100, 20)	0.724 637	0.754 296	<b>0.725 392</b>	<b>0.754 389</b>
(200, 10)	0.743 741	0.916 359	<b>0.746 812</b>	<b>0.931 341</b>
(200, 20)	0.711 006	0.742 119	<b>0.712 502</b>	<b>0.744 224</b>
(500, 20)	<b>0.704 720</b>	0.747 942	0.704 430	<b>0.749 465</b>

由表1可知:对于平均值,在实例(500, 20)中,HMOEA/D+Random略优于HMOEA/D+HI,在其他实例中,HMOEA/D+HI均优于HMOEA/D+Random;对于最好值,HMOEA/D+HI均优于HMOEA/D+Random.综上所述,所提出的3种初始化策略在求解GPFSP上表现更优。

### 3.5 禁忌搜索策略的有效性

为了验证禁忌搜索策略的有效性,本文编码两种算法,一种是不采用禁忌搜索策略的HMOEA/DNTS,另外一种结合禁忌搜索策略的HMOEA/D+TS,其他所有参数设置相同,在所有测试数据集上独立运行20次并统计HV的平均值和最好值,结果如表2所示。

表2 HMOEA/DNTS和HMOEA/D+TS对比结果

实例	HMOEA/DNTS		HMOEA/D+TS	
	平均值	最好值	平均值	最好值
(20, 5)	0.714 100	0.824 157	<b>0.862 299</b>	<b>0.975 062</b>
(20, 10)	0.765 283	0.812 174	<b>0.881 929</b>	<b>0.932 634</b>
(20, 20)	0.702 970	0.733 611	<b>0.852 752</b>	<b>0.876 037</b>
(50, 5)	0.683 607	0.753 010	<b>0.853 994</b>	<b>0.888 585</b>
(50, 10)	0.685 231	0.742 866	<b>0.910 228</b>	<b>0.946 232</b>
(50, 20)	0.582 527	0.646 742	<b>0.870 255</b>	<b>0.906 698</b>
(100, 5)	0.685 253	0.752 204	<b>0.902 996</b>	<b>0.964 768</b>
(100, 10)	0.625 356	0.655 632	<b>0.882 344</b>	<b>0.932 560</b>
(100, 20)	0.518 444	0.560 665	<b>0.891 175</b>	<b>0.904 344</b>
(200, 10)	0.614 027	0.668 761	<b>0.917 880</b>	<b>0.987 527</b>
(200, 20)	0.456 652	0.539 774	<b>0.910 381</b>	<b>0.926 124</b>
(500, 20)	0.425 025	0.459 293	<b>0.938 851</b>	<b>0.961 012</b>

由表2结果可知:所提出的禁忌搜索策略在求解GPFSP上表现更优。

### 3.6 节能策略的有效性

为了验证节能策略的有效性,本文编码两种算法,一种是不采用节能策略的HMOEA/DNES,另外一种结合节能策略的HMOEA/D+ES,其他所有参数设置相同,在所有测试数据集上独立运行20次并统计HV的平均值和最好值,结果如表3所示。

表3 HMOEA/DNES和HMOEA/D+ES对比结果

实例	HMOEA/DNES		HMOEA/D+ES	
	平均值	最好值	平均值	最好值
(20, 5)	0.637 523	0.860 884	<b>0.803 405</b>	<b>0.944 907</b>
(20, 10)	0.611 102	0.748 151	<b>0.833 860</b>	<b>0.893 739</b>
(20, 20)	0.504 211	0.595 534	<b>0.778 023</b>	<b>0.834 966</b>
(50, 5)	0.389 733	0.550 021	<b>0.710 243</b>	<b>0.858 399</b>
(50, 10)	0.563 402	0.795 271	<b>0.780 757</b>	<b>0.890 115</b>
(50, 20)	0.481 340	0.594 986	<b>0.717 934</b>	<b>0.759 329</b>
(100, 5)	0.510 239	<b>0.843 920</b>	<b>0.735 538</b>	0.816 602
(100, 10)	0.397 782	0.588 238	<b>0.721 507</b>	<b>0.781 217</b>
(100, 20)	0.429 493	0.562 537	<b>0.667 978</b>	<b>0.700 491</b>
(200, 10)	0.471 359	0.807 925	<b>0.721 980</b>	<b>0.870 042</b>
(200, 20)	0.383 329	0.446 982	<b>0.633 472</b>	<b>0.654 937</b>
(500, 20)	0.232 516	0.331 485	<b>0.612 600</b>	<b>0.678 683</b>

由表3可知:对于最好值,在实例(100, 5)中,HMOEA/DNES略优于HMOEA/D+ES,而在其他实例中HMOEA/D+ES均优于HMOEA/DNES;对于平均值,HMOEA/D+ES均优于HMOEA/DNES.综上所述,所提出的节能策略在求解GPFSP上表现更优。

### 3.7 对比实验

为了进一步验证HMOEA/D求解GPFSP的有效性,将HMOEA/D与当前主流的多目标进化算法进行比较,其中包括:MOEA/D、NSGA-II、AdaW<sup>[32]</sup>和AMEA<sup>[34]</sup>. AdaW在进化过程中通过自适应调整其权重向量引导种群向更好的方向收敛;AMEA采用4种协同搜索算子,并根据性能自适应地调整它们的利用率,以此探索更优解.为了确保公平性,所有算法都在测试数据集上独立运行20次,所有算法均设置100 000次评价次数,统计每个测试数据集上HV的平均值和最好值,结果如表4所示。

由表4可知:在相同评价次数下,HMOEA/D的HV指标的平均值和最好值在大部分测试集上均显著优于其他对比算法,在测试集(50, 20)和测试集(100, 20)上,AMEA略优于HMOEA/D,但其性能非常接近,由此验证了HMOEA/D求解GPFSP的有效性。

表 4 100 000次评价次数下5个算法的HV值对比结果

实例	HMOEA/D		MOEA/D		NSGA-II		AdaW		AMEA	
	平均值	最好值	平均值	最好值	平均值	最好值	平均值	最好值	平均值	最好值
(20, 5)	<b>0.801 364</b>	<b>0.879 394</b>	0.533 565	0.633 833	0.544 804	0.619 869	0.313 868	0.576 034	0.606 939	0.677 638
(20, 10)	<b>0.744 352</b>	<b>0.771 352</b>	0.504 193	0.570 620	0.589 785	0.647 325	0.416 334	0.554 318	0.608 865	0.640 736
(20, 20)	<b>0.761 516</b>	<b>0.812 539</b>	0.418 792	0.460 448	0.514 309	0.541 408	0.401 469	0.460 929	0.469 137	0.548 302
(50, 5)	<b>0.869 089</b>	<b>0.894 408</b>	0.267 707	0.325 294	0.301 265	0.310 682	0.166 598	0.325 058	0.297 741	0.332 639
(50, 10)	<b>0.710 627</b>	<b>0.757 327</b>	0.414 982	0.559 271	0.618 538	0.660 403	0.312 514	0.515 326	0.577 773	0.687 163
(50, 20)	0.604 884	0.644 686	0.347 404	0.420 182	0.563 685	0.591 985	0.272 464	0.370 737	<b>0.621 390</b>	<b>0.656 726</b>
(100, 5)	<b>0.793 380</b>	<b>0.859 799</b>	0.399 300	0.495 007	0.526 938	0.555 528	0.169 497	0.490 590	0.528 106	0.585 027
(100, 10)	<b>0.677 622</b>	<b>0.719 383</b>	0.284 177	0.368 696	0.453 588	0.501 755	0.200 162	0.323 178	0.510 819	0.531 433
(100, 20)	0.583 856	0.635 431	0.264 415	0.379 640	0.548 692	0.578 151	0.213 926	0.326 877	<b>0.622 451</b>	<b>0.683 860</b>
(200, 10)	<b>0.731 829</b>	<b>0.771 938</b>	0.192 389	0.311 667	0.461 583	0.480 786	0.066 344	0.289 523	0.451 935	0.512 104
(200, 20)	<b>0.565 564</b>	<b>0.595 094</b>	0.152 528	0.225 706	0.471 856	0.498 716	0.109 267	0.213 967	0.528 296	0.588 231
(500, 20)	<b>0.629 032</b>	<b>0.695 489</b>	0.091 744	0.192 596	0.517 075	0.552 405	0.041 298	0.102 005	0.575 510	0.622 083

表 5 展示了 5 个算法的 Friedman 秩和检验, HMOEA/D 排名第 1, 说明 HMOEA/D 获得的帕累托前沿具有更大的 HV 值. 此外, 所有算法的  $p$  值小于显著性水平 0.05, 因此 5 个算法具有明显差异. 基于上述结果和分析可以得出结论: HMOEA/D 对比其他算法具有更好的性能.

表 5 Friedman秩和检验

算法	排序	$p$ 值
HMOEA/D	5.00	1.34e-09
MOEA/D	2.00	
NSGA-II	3.08	
AdaW	1.00	
AMEA	3.92	

表 6 展示了 5 个算法在所有实例上的运行时间, 由此可知: MOEA/D 算法在所有实例上消耗的时间最少; NSGA-II 算法在实例 (20, 5), (20, 10), (20, 20) 上消耗的时间最多, 在其他实例上, HMOEA/D 算法消耗的时间最多. 尽管 HMOEA/D 算法在大多数实例上消耗的时间最多, 但仍在可接受的范围内, 并且 HMOEA/D 算法的性能均优于其他算法, 通过消耗时间得到更好的解是非常值得的. 综上, HMOEA/D 算法对比其他算法具有良好的收敛性和稳定性.

表 6 5个算法独立运行20次的消耗时间对比结果

	HMOEA/D	MOEA/D	NSGA-II	AdaW	AMEA
(20, 5)	451.34	<b>58.94</b>	889.53	92.24	219.07
(20, 10)	459.12	<b>34.91</b>	731.98	90.57	204.52
(20, 20)	492.47	<b>40.69</b>	801.68	91.85	225.21
(50, 5)	1 406.91	<b>93.48</b>	940.33	100.30	390.88
(50, 10)	1 602.52	<b>33.23</b>	777.24	90.61	434.60
(50, 20)	1 648.02	<b>36.57</b>	697.41	95.79	523.52
(100, 5)	3 617.46	<b>33.22</b>	888.34	92.58	760.29
(100, 10)	3 895.87	<b>36.16</b>	746.31	91.65	926.81
(100, 20)	4 509.40	<b>39.21</b>	723.58	98.42	1 229.98
(200, 10)	12 361.87	<b>39.17</b>	787.31	97.74	2 452.45
(200, 20)	14 534.81	<b>47.72</b>	700.23	111.78	3 679.62
(500, 20)	36 925.12	<b>78.53</b>	731.69	144.22	23 790.89

### 4 结 论

为了降低生产过程中的能量消耗, 优化最大完工时间, 本文提出了混合 MOEA/D 算法求解 GPFSP. 该算法结合 MOEA/D 较强的优化性能, 通过混合初始化策略获得高质量的初始解, 并采用禁忌搜索替代局部搜索跳出局部最优, 进而充分探索解空间. 同时本文还设计了一种新颖的节能策略, 在不增加最大完工时间目标的基础上进一步优化总能量消耗目标. 为了验证算法性能, 将其与其他相近算法进行对比, 帕累托前沿对比图表明, HMOEA/D 算法可以得到最优的帕累托前沿, 其收敛性和分布性更好, 箱线图对比图表明, HMOEA/D 算法得到的箱线图的宽度更窄, 其稳定性更高.

后续工作将考虑更多实际问题约束, 将目标函数扩展, 结合强化学习相关技术求解多目标置换流水车间调度问题. 可以更多地考虑分布式置换流水调度问题, 结合特定的动态事件, 使其更为贴合实际的生产环境.

### 参考文献(References)

[1] 王凌, 潘子肖. 基于深度强化学习与迭代贪婪的流水车间调度优化[J]. 控制与决策, 2021, 36(11): 2609-2617.  
(Wang L, Pan Z X. Scheduling optimization for flow-shop based on deep reinforcement learning and iterative greedy method[J]. Control and Decision, 2021, 36(11): 2609-2617.)

[2] 王凌, 郑洁, 王晶晶. 求解区间数分布式流水线调度的混合离散果蝇优化算法[J]. 控制与决策, 2020, 35(4): 930-936.  
(Wang L, Zheng J, Wang J J. A hybrid discrete fruit fly optimization algorithm for distributed permutation flowshop scheduling with interval data[J]. Control and Decision, 2020, 35(4): 930-936.)

- [3] 郑小操, 龚文引. 改进人工蜂群算法求解模糊柔性作业车间调度问题[J]. 控制理论与应用, 2020, 37(6): 1284-1292.  
(Zheng X C, Gong W Y. An improved artificial bee colony algorithm for fuzzy flexible job-shop scheduling problem[J]. Control Theory & Applications, 2020, 37(6): 1284-1292.)
- [4] 王凌, 王晶晶. 带峰值能耗约束流水线调度的协同群智能优化[J]. 控制与决策, 2021, 36(10): 2350-2358.  
(Wang L, Wang J J. Cooperative memetic optimization for flowshop scheduling with peak power consumption constraint[J]. Control and Decision, 2021, 36(10): 2350-2358.)
- [5] Li Y, Li X Y, Gao L. An effective solution space clipping-based algorithm for large-scale permutation flow shop scheduling problem[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2023, 53(1): 635-646.
- [6] de Fátima Morais M, Ribeiro M H D M, da Silva R G, et al. Discrete differential evolution metaheuristics for permutation flow shop scheduling problems[J]. Computers & Industrial Engineering, 2022, 166: 107956.
- [7] 张梓琪, 钱斌, 胡蓉, 等. 基于多维EDA算法的低碳分布式装配流水线车间调度[J]. 控制与决策, 2022, 37(5): 1367-1377.  
(Zhang Z Q, Qian B, Hu R, et al. Multidimensional estimation of distribution algorithm for low carbon scheduling of distributed assembly permutation flow-shop[J]. Control and Decision, 2022, 37(5): 1367-1377.)
- [8] Shao Z S, Pi D C, Shao W S, et al. An efficient discrete invasive weed optimization for blocking flow-shop scheduling problem[J]. Engineering Applications of Artificial Intelligence, 2019, 78: 124-141.
- [9] Lu C, Huang Y X, Meng L L, et al. A Pareto-based collaborative multi-objective optimization algorithm for energy-efficient scheduling of distributed permutation flow-shop with limited buffers[J]. Robotics and Computer-Integrated Manufacturing, 2022, 74: 102277.
- [10] Zhao F Q, Ma R, Wang L. A self-learning discrete jaya algorithm for multiobjective energy-efficient distributed no-idle flow-shop scheduling problem in heterogeneous factory system[J]. IEEE Transactions on Cybernetics, 2022, 52(12): 12675-12686.
- [11] Wang J J, Wang L. A knowledge-based cooperative algorithm for energy-efficient scheduling of distributed flow-shop[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(5): 1805-1819.
- [12] Lu C, Gao L, Gong W Y, et al. Sustainable scheduling of distributed permutation flow-shop with non-identical factory using a knowledge-based multi-objective memetic optimization algorithm[J]. Swarm and Evolutionary Computation, 2021, 60: 100803.
- [13] Pan Q K, Gao L, Wang L. An effective cooperative co-evolutionary algorithm for distributed flowshop group scheduling problems[J]. IEEE Transactions on Cybernetics, 2022, 52(7): 5999-6012.
- [14] He X, Pan Q K, Gao L, et al. A greedy cooperative co-evolutionary algorithm with problem-specific knowledge for multiobjective flowshop group scheduling problems[J]. IEEE Transactions on Evolutionary Computation, 2023, 27(3): 430-444.
- [15] Tian S S, Li X Y, Wan J, et al. A novel cuckoo search algorithm for solving permutation flowshop scheduling problems[C]. 2021 IEEE International Conference on Recent Advances in Systems Science and Engineering. Shanghai, 2022: 1-8.
- [16] Khurshid B, Maqsood S, Omair M, et al. An improved evolution strategy hybridization with simulated annealing for permutation flow shop scheduling problems[J]. IEEE Access, 2021, 9: 94505-94522.
- [17] Ouchiekh R, Fri M, Touil A, et al. Total weighted tardiness in the permutation flow shop under uncertainty[J]. IFAC-PapersOnLine, 2021, 54(1): 1174-1180.
- [18] Ni Z B, Wang B, Wu B. A hybrid algorithm for a robust permutation flow shop scheduling problem[C]. 2020 Chinese Control and Decision Conference. Hefei, 2020: 3802-3807.
- [19] Pan Z X, Wang L, Wang J J, et al. Deep reinforcement learning based optimization algorithm for permutation flow-shop scheduling[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2023, 7(4): 983-994.
- [20] Pavelski L M, Kessaci M É, Delgado M. Local optima network sampling for permutation flowshop[C]. 2021 IEEE Congress on Evolutionary Computation (CEC). Kraków, 2021: 1131-1138.
- [21] Lu C, Gao L, Yi J, et al. Energy-efficient scheduling of distributed flow shop with heterogeneous factories: A real-world case from automobile industry in China[J]. IEEE Transactions on Industrial Informatics, 2021, 17(10): 6687-6696.
- [22] 李瑞, 王凌, 龚文引. 知识驱动的模式算法求解分布式绿色柔性调度[J]. 华中科技大学学报: 自然科学版, 2022, 50(6): 55-60.  
(Li R, Wang L, Gong W Y. Knowledge-driven memetic algorithm for distributed green flexible job shop scheduling problem[J]. Journal of Huazhong University of Science and Technology: Nature Science Edition, 2022, 50(6): 55-60.)
- [23] 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.)
- [24] 李瑞, 龚文引. 改进的基于分解的多目标进化算法求解双目标模糊柔性作业车间调度问题[J]. 控制理论与应用, 2022, 39(1): 31-40.  
(Li R, Gong W Y. An improved multi-objective evolutionary algorithm based on decomposition for bi-objective fuzzy flexible job-shop scheduling problem[J]. Control Theory & Applications, 2022, 39(1): 31-40.)
- [25] Jiang E D, Wang L. An improved multi-objective evolutionary algorithm based on decomposition for energy-efficient permutation flow shop scheduling problem with sequence-dependent setup time[J]. International Journal of Production Research, 2019, 57(6): 1756-1771.
- [26] Rossit D G, Nesmachnow S, Rossit D A. A multi objective evolutionary algorithm based on decomposition for a flow shop scheduling problem in the context of industry 4.0[J]. International Journal of Mathematical, Engineering and Management Sciences, 2022, 7(4): 433-454.
- [27] Yang J Q, Yang S Y, Ni P H. A vector tabu search algorithm with enhanced searching ability for Pareto solutions and its application to multiobjective optimizations[J]. IEEE Transactions on Magnetics, 2016, 52(3): 1-4.
- [28] Li X Y, Lu C, Gao L, et al. An effective multiobjective algorithm for energy-efficient scheduling in a real-life welding shop[J]. IEEE Transactions on Industrial Informatics, 2018, 14(12): 5400-5409.
- [29] Nawaz M, Ensco E E, Ham I. A heuristic algorithm for the  $m$ -machine,  $n$ -job flow-shop sequencing problem[J]. Omega, 1983, 11(1): 91-95.
- [30] Gao K Z, Suganthan P N, Chua T J, et al. A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion[J]. Expert Systems With Applications, 2015, 42(21): 7652-7663.
- [31] Zhang Q F, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.
- [32] Li M Q, Yao X. What weights work for You? Adapting weights for any Pareto front shape in decomposition-based evolutionary multiobjective optimisation[J]. Evolutionary Computation, 2020, 28(2): 227-253.
- [33] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [34] Pan Z X, Wang L, Chen J F, et al. A novel evolutionary algorithm with adaptation mechanism for fuzzy permutation flow-shop scheduling[C]. 2021 IEEE Congress on Evolutionary Computation. Kraków, 2021: 367-374.
- [35] While L, Hingston P, Barone L, et al. A faster algorithm for calculating hypervolume[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(1): 29-38.
- [36] Taillard E. Benchmarks for basic scheduling problems[J]. European Journal of Operational Research, 1993, 64(2): 278-285.
- [37] Roy R K. Design of experiments using the Taguchi approach: 16 steps to product and process improvement[M]. New York: John Wiley, 2001: 479-482.

### 作者简介

罗聪(2000—), 男, 硕士生, 从事智能优化与应用的研究, E-mail: 1202121530@cug.edu.cn;

龚文引(1979—), 男, 教授, 博士生导师, 从事智能优化与应用、车间调度等研究, E-mail: wygong@cug.edu.cn.