# 具有双资源约束的分布式装配混合流水车间调度

赵 才[1] 吴亮红[2*]
1.湖南科技大学机电工程学院,湘潭,411100
2.湖南科技大学信息与电气工程学院,湘潭,411100

**摘要**:针对具有双资源约束的分布式装配混合流水车间调度问题,提出一种基于知识驱动的迭代贪婪(KDIG)算法。采用基于知识的NEH(Nawaz-Enscore-Ham)初始化策略,生成初始解决方案。基于问题特征设计了4种局部搜索算子。局部搜索算子与Q学习机制的结合使个体在迭代更新时动态选择最优的局部搜索算子,从而显著提升个体在搜索过程中的效率。KDIG算法与5种主流算法在81个大型实例上的测试结果表明,KDIG算法优于其他对比算法。

**关键词**:分布式装配混合流车间调度;知识驱动;迭代贪婪算法;Q学习
**中图分类号**:TP182
DOI:10.3969/j.issn.1004-132X.2026.07.022 **开放科学(资源服务)标识码(OSID)**:

# Distributed Assembly Hybrid Flow Shop Scheduling with Dual Resource Constraints

ZHAO Cai[1] WU Lianghong[2*]
1.School of Mechanical Engineering, Hunan University of Science and Technology, Xiangtan, Hunan, 411100
2.School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, Hunan, 411100

**Abstract**: This paper proposes a knowledge-driven iterative greedy (KDIG) algorithm for the distributed assembly hybrid flow shop scheduling problem with dual resource constraints. The algorithm adopts a knowledge-based NEH (Nawaz-Enscore-Ham) initialization strategy to generate the initial solution. Based on the problem characteristics, four local search operators are designed. Combined with a Q-learning mechanism, these operators enable individuals to dynamically select the optimal local search operator during iterative updates, thereby significantly improving search efficiency. Experimental results on 81 large-scale instances, comparing the KDIG algorithm with five other mainstream algorithms, demonstrate that the proposed KDIG algorithm outperforms all benchmark algorithms.

**Key words**: distributed assembly hybrid flow shop scheduling; knowledge-driven; iterative greedy algorithm; Q-learning

## 0 引言

以提高生产灵活性并降低大规模定制复杂性为目标的分布式制造已成为一种流行的生产模式[1]。分布式混合流水车间调度已在汽车、钢铁、电子、半导体、纺织和食品等行业应用广泛,相关研究倍受关注。ZHANG等[2]提出一种结合粒子群优化和Q-learning局部搜索的多目标模因算法。DI等[3]提出一种多智能体强化学习方法来最小化完工时间和总能耗。LIU等[4]针对具有阻塞约束的分布式混合流水车间调度问题(distributed hybrid flow shop scheduling problem,DHFSSP),提出一种改进迭代贪婪算法。CUI等[5]研究了具有不相关并行机器的多目标DHFSSP,提出一种改进的多种群遗传算法。SHAO等[6]提出一种能量和劳动感知的网络模因算法解决DHFSSP。

虽然DHFSSP的研究已取得丰硕成果,但大多数研究聚焦于机器资源的优化配置。生产调度不仅受限于机器资源,还需综合考虑人力资源配置。不同客户在产品规格、订单批量、交货周期等方面的定制化需求也应满足。这使得生产过程必须灵活调整以适应每个生产订单的独有特征。因此,研究DHFSSP必须全面考虑复杂的生产场景[7-9]和生产资源的约束条件[10-11]。

工作未能在最后期限内完成不仅会引发合同

**收稿日期**:2024-11-27 **修回日期**:2026-04-06
**基金项目**:国家重点研发计划(2023YFC3011100);国家自然科学基金(62373146);湖南省自然科学基金(2022JJ30265,2023JJ40286);湖南省人才托举工程青年人才项目(2022TJ-Q03).

处罚，还会造成声誉损害，故研究具有生产装配与工人资源约束，以最小化总延迟（total tardiness，TTD）为目标的具有双资源约束的分布式装配混合流车间调度问题（distributed assembly hybrid flow shop scheduling problem with dual resource constraints，DAHFSSP-DRC）很有意义。

迭代贪婪（iterative greedy，IG）算法算法是一种高效简单的算法，起初用于解决求解流水车间调度问题[12]，后被广泛用于解决各类调度问题。JING等[13]提出不确定处理时间的分布式置换流水车间调度问题，并通过IG算法进行求解。ZOU等[14]采用自适应IG算法解决制造车间中的自动引导车辆调度问题。为处理阻塞混合流水车间问题，QIN等[15]提出利用交换策略的IG算法。ZHAO等[16]利用基于种群的IG算法解决了分布式装配无等待流水车间调度问题，通过新的构造启发式方法初始化起始种群。对于分布式阻塞流水车间调度问题，KARIMI-MAMAGHAN等[17]在搜索过程中加入Q-Learning来选择合适的扰动算子，提出一种基于Q-Learning的IG算法来解决置换流水车间调度问题。

知识驱动的迭代贪婪（knowledge-driven iterative greedy，KDIG）算法被提出并用于求解DAHFSSP-DRC。本文采用基于知识的NEH（Nawaz-Enscore-Ham）初始化策略生成初始解决方案。结合4种局部搜索算子与Q-learning机制，使个体在迭代更新时动态选择最优的局部搜索算子。81个大型实例的算法对比验证了KDIG算法的优越性。

## 1 问题描述和数学模型

### 1.1 DAHFSSP-DRC的描述

根据文献[18]，DAHFSSP-DRC具体描述如下：每个产品的生产分为加工和装配。加工在具有双资源约束的分布式混合流水车间内完成。每个工件都在提前指定的工厂内加工。工件一旦分配到某个工厂，就不能转移到其他工厂。具有相同生产流程的$N$个工件有连续的$N_S$个加工阶段。每个阶段由一组机器$K_{i,j,f}$和能加工工件$O_{i,j}$的一组工人$W_{i,j,f}$组成。工件$O_{i,j}$的加工时间提前确定，且不受加工工人的影响。工件$O_{i,j}$加工完毕后，进入装配阶段，即在有一台机器的装配车间逐个组装每个产品。本文用到的量符号及其含义如表1所示。

### 1.2 数学模型

DAHFSSP-DRC的数学模型如下：

**表1 文中量符号的含义**

**Tab.1 List of symbols used in the paper**

| 量符号 | 说明 |
|---|---|
| $i,i'$ | 工件索引，$i,i'=1,2,\cdots,N$ |
| $j$ | 阶段的索引，$j=1,2,\cdots,N_S$ |
| $k$ | 机器的索引，$k=1,2,\cdots,N_M$ |
| $w$ | 工人索引，$w=1,2,\cdots,N_W$ |
| $f$ | 工厂索引，$f=1,2,\cdots,N_F$ |
| $p,p'$ | 产品索引，$p,p'=1,2,\cdots,N_P$ |
| $I$ | 工件集合 |
| $I_p$ | 产品$p$的工件集合 |
| $S$ | 阶段集合 |
| $P$ | 产品集合 |
| $F$ | 工厂集合 |
| $K_f$ | 工厂$f$中机器的集合 |
| $K_{i,j,f}$ | 工厂$f$中工件$O_{i,j}$的加工机器集合 |
| $W_f$ | 工厂$f$中工人的集合 |
| $W_{i,j,f}$ | 工厂$f$中工件$O_{i,j}$的加工工人集合 |
| $J_i$ | 第$i$个工件 |
| $O_{i,j}$ | 第$j$阶段的工件$i$ |
| $F_f$ | 第$f$个工厂 |
| $M_{j,k}$ | 第$j$阶段的第$k$台机器 |
| $W_{j,w}$ | 第$j$阶段的第$w$个工人 |
| $C_{i,j}$ | $O_{i,j}$的完工时间 |
| $N_F$ | 工厂总数 |
| $N_P$ | 产品总数 |
| $N_M$ | 工厂中的机器总数 |
| $N_{M,j}$ | 第$j$阶段的机器总数 |
| $N_W$ | 工人总数 |
| $N_{W,j}$ | 第$j$阶段的工人总数 |
| $T_p$ | 产品$p$的延迟时间 |
| $T_{i,j,f,k,w}$ | 工件的实际加工时间 |
| $T_p$ | 产品$p$的装配时间 |
| $D_p$ | 产品$p$的离开时间 |
| $G_{i,p}$ | 工件$i$属于产品$p$ |
| $C_p$ | 产品$p$的完工时间 |
| $S_p$ | 产品$p$的开始时间 |
| $L$ | 无穷大的正数 |
| $S_{i,j}$ | $O_{i,j}$的开始时间 |
| $T_{TD}$ | 总延迟 |
| $V_{i,j,f,k,w,s}$ | 若工件由工人$w$处理，则为1，否则为0 |
| $X_{i,i',j,f,k}$ | 若机器在工件$i$后加工$i'$，则为1，否则为0 |
| $Y_{i,i',j,f,k}$ | 若工人在工件$i$后加工$i'$，则为1，否则为0 |
| $Z_{i,f}$ | 若工件$i$在工厂$f$中加工，则为1，否则为0 |
| $H_{p,p'}$ | 产品$p$后紧跟$p'$产品，则为1，否则为0 |

$$\min T_{TD}=\sum_{p\in P}T_p \tag{1}$$

$$T_p=\max\{C_p-D_p,0\}\quad p\in P \tag{2}$$

$$\sum_{f\in F}Z_{i,f}=1\quad \forall i\in I \tag{3}$$

$$Z_{i,f}=\sum_{k\in K_{i,j,f}}\sum_{w\in W_{i,j,f}}V_{i,j,f,k,w}\quad \forall i\in I,j\in S,f\in F \tag{4}$$

$$Z_{i,f}=\sum_{i'\in I-i}\sum_{k\in K_{i,j,f}}X_{i,i',j,f,k}\quad\forall i\in I,j\in S,f\in F \tag{5}$$

$$Z_{i,f}=\sum_{i'\in I-i}\sum_{w\in W_{i,j,f}}Y_{i,i',j,f,w}\quad\forall i\in I,j\in S,f\in F \tag{6}$$

$$\sum_{i'\in I-i}\sum_{f\in F}\sum_{k\in K_{i,j,f}}X_{i,i',j,f,k}=1\quad\forall i\in I,j\in S \tag{7}$$

$$\sum_{i'\in I-i}\sum_{f\in F}\sum_{k\in K_{i,j,f}}Y_{i,i',j,f,k}=1\quad\forall i\in I,j\in S \tag{8}$$

$$S_{i,j}+\sum_{f\in F}\sum_{k\in K_{i,j,f}}\sum_{w\in W_{i,j,f}}PT_{i,j,f,k,w}V_{i,j,f,k,w}\leqslant S_{i,j+1}\quad\forall i\in I,j\in S \tag{9}$$

$$C_{i',j}-\sum_{f\in F}\sum_{k\in K_{i,j,f}}\sum_{w\in W_{i,j,f}}PT_{i,j,k,w,f}V_{i,j,k,w,f}+L(1-\sum_{f\in F}\sum_{k\in K_{i,j,f}}X_{i,i',j,f,k})\geqslant C_{i,j}\quad\forall i\in I,i'\in I,j\in S \tag{10}$$

$$C_{i',j}-\sum_{f\in F}\sum_{k\in K_{i,j,f}}\sum_{w\in W_{i,j,f}}PT_{i,j,k,w,f}V_{i,j,k,w,f}+L(1-\sum_{f\in F}\sum_{k\in K_{i,j,f}}Y_{i,i',j,f,k})\geqslant C_{i,j}\quad\forall i\in I,i'\in I,j\in S \tag{11}$$

$$\sum_{i'\in I-i}X_{i,i',j,f,k}=\sum_{i'\in I-i}X_{i',i,j,f,k}\quad\forall i\in I,j\in S,f\in F,k\in K_{i,j,f} \tag{12}$$

$$\sum_{i'\in I-i}Y_{i,i',j,f,w}=\sum_{i'\in I-i}Y_{i',i,j,f,k}\quad\forall i\in I,j\in S,f\in F,w\in W_{i,j,f} \tag{13}$$

$$\sum_{i'\in I-i}X_{i,i',j,f,k}=\sum_{w\in W_{i,j,f}}V_{i,j,f,k,w}\quad\forall i\in I,j\in S,f\in F,k\in K_{i,j,f} \tag{14}$$

$$\sum_{i'\in I-i}Y_{i,i',j,f,w}=\sum_{k\in K_{i,j,f}}V_{i,j,f,k,w}\quad\forall i\in I,j\in S,f\in F,w\in W_{i,j,f} \tag{15}$$

$$\sum_{P'\in P-p}H_{p,p'}=1\quad\forall p\in P \tag{16}$$

$$\sum_{P\in P-p'}H_{p,p'}=1\quad\forall p'\in P \tag{17}$$

$$H_{p,p'}+H_{p',p}\leqslant 1\quad\forall p\in P,p\in P \tag{18}$$

$$C_p\geqslant C_{i,N^s}G_{i,p}+AT_p\quad\forall i\in I,p\in P \tag{19}$$

$$C_{p'}\geqslant C_p+AT_{p'}+L(1-H_{p,p'})\quad\forall i\in I,p\in P,p'\in P \tag{20}$$

$$S_{i,j}\geqslant 0,C_{i,j}\geqslant 0\quad\forall i\in I,j\in J \tag{21}$$

$$S_p\geqslant 0,C_p\geqslant 0\quad\forall p\in P \tag{22}$$

式(3)为工件绑定约束，确保每个工件严格在指定工厂加工；式(4)～式(6)为工序完整性约束，确保每个工件所有加工阶段在同一个工厂连续加工完成；式(7)、式(8)为单资源分配约束，限定每个工件的所有工序阶段只能由单一工人—机器组成；式(9)表示所有阶段的每个工件需按照给定顺序加工；式(10)为机器冲突规避约束，保证同时段单台机器仅加工一个工件；式(11)保证一个工人一次仅加工一个工件；式(12)、式(13)确保机器中有紧前和紧后的工件，防止出现加工时间浪费；式(15)确保任何时间段的工人均只能加工一个工件；式(16)～式(18)为产品时序约束，维护装配线产品的先后关系并保证每个产品都有紧前和紧后的产品；式(19)表示装配触发条件，要求产品所有组件完工后方可启动装配；式(20)保证任意两个产品之间的装配时间不重叠。

### 1.3　DAHFSSP-DRC 问题的知识

引用文献[19]提出的两种知识设计 KDIG 算法。

知识 1(K1)：将属于同一产品的工件分配到不同的工厂，有助于提前产品的开始装配时间。

知识 2(K2)：如果属于同一产品的工件在每个工厂中一起分配，则该产品的组装可更早开始。

## 2　KDIG 算法求解 DAHFSSP-DRC

IG 算法是一种简单的迭代搜索方法，易于编码和理解。本文以 IG 算法为主体框架，并结合问题知识设计高效的策略，从而提出 KDIG 算法。KDIG 算法主要由基于知识的初始化策略、破坏和重构，以及基于 Q-learning 的变邻域局部搜索、验收准则组成。图 1 给出 KDIG 算法的主要流程。

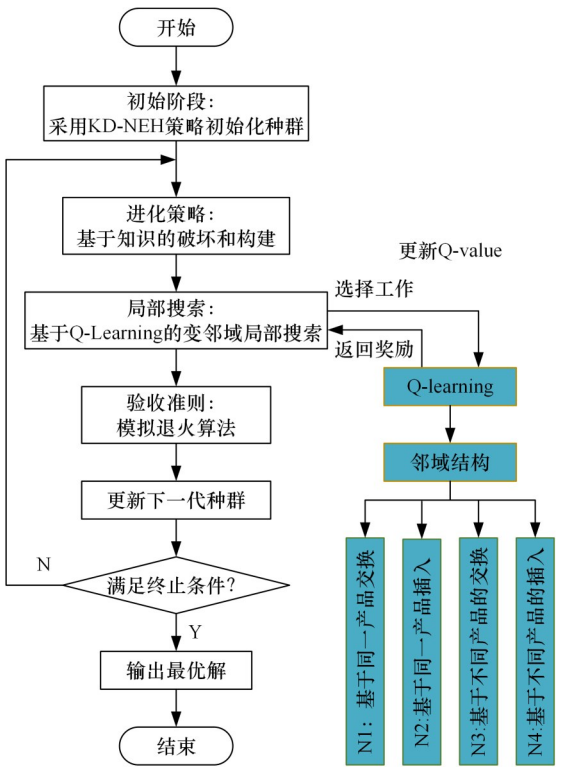



**图 1　KDIG 算法的流程图**

**Fig.1　Flowchart of KDIG algorithm**

### 2.1　编码方式

DAHFSSP-DRC 的加工部分包含 4 个子问题：①工件的工厂分配；②工件的加工顺序；③工件的机器分配；④工人分配。组装阶段只涉及产品的装配顺序。采用完整编码结构即每个编码对应一个子问题导致解空间过大。因此，采用部分编码结构表示每个解[18]。具体而言，编码由 $N_F$ 个

部分组成，每个部分对应一个工厂并包含表示该工厂在加工阶段的工件加工顺序。$\boldsymbol{\pi}=(\boldsymbol{\pi}_1,\boldsymbol{\pi}_2,\cdots,\boldsymbol{\pi}_{N_F})$为DAHFSSP-DRC的一个可行解，其中，$\pi_f$为加工阶段分配给工厂$f$的部分工件加工编码（排序见图2），其中，$F_1$、$F_2$分别表示工厂1和工厂2。虽然$\boldsymbol{\pi}$只能表示所有工件在加工阶段的工厂分配和工件加工排序，但能降低解空间的复杂度，避免无效搜索。

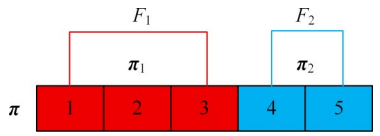



**图2 编码示例**

**Fig.2 An example of encoding**

## 2.2 解码方式

优秀的解码可保证解的质量，将解码分为加工和装配两部分。在加工部分，每个阶段所有工件的加工顺序采用最早完工时间规则[20]。在第一阶段，所有工件都按照解决方案$\boldsymbol{\pi}$中的顺序加工。在后续阶段，所有工件将根据上一阶段的完成时间按升序排序。每个阶段采用机器负载平衡规则分配机器，采用工人负载平衡规则分配工人。每个阶段的工件采用左移解码策略对解进行解码[18]。通过这种策略，能尽早加工满足机器约束和工人约束的工件。左移解码策略解码的步骤如下：

1）根据最早完工时间规则获取每个阶段的工件加工顺序后，根据加工顺序对所有工件采用基于机器负载平衡规则，对所有工人采用工人负载平衡规则，得到加工机器$k$和工人$w$。

2）在机器$k$和工人$w$上从左到右依次求出空闲时间，然后根据下式

$$\left.\begin{aligned}&\max\{B_{f,k}^{(u_1)},B_{f,w}^{(u_2)},0\}+T_{i,j,f,k,w}\leqslant\\&\quad\min\{E_{f,k}^{(u_1)},E_{f,w}^{(u_2)}\}\qquad j=1\\&\max\{B_{f,k}^{(u_1)},B_{f,w}^{(u_2)},C_{i,j-1}\}+T_{i,j,f,k,w}\leqslant\\&\quad\min\{E_{f,k}^{(u_1)},E_{f,w}^{(u_2)}\}\qquad j=2,3,\cdots\end{aligned}\right\}\tag{23}$$

式中：$B_{f,k}^{(u_1)}$、$E_{f,k}^{(u_1)}$分别为工厂$f$中机器$k$的第$u_1$个空闲时段的开始时间和结束时间；$B_{f,w}^{(u_2)}$、$E_{f,w}^{(u_2)}$分别为工厂$f$中工人$w$的第$u_2$个空闲时段的开始时间和结束时间。

确定判断是否可以将工件$O_{i,j}$插入空闲。因此，工件$O_{i,j}$的加工起始时间为

$$S_{i,j}=\begin{cases}\max\{B_{f,k}^{(u_1')},B_{f,w}^{(u_2')},0\} & j=1\\ \max\{B_{f,k}^{(u_1')},B_{f,w}^{(u_2')},C_{i,j-1}\} & j=2,3,\cdots\end{cases}\tag{24}$$

式中：$u_1'$、$u_2'$分别为机器$k$和工人$w$的第一个空闲时段，并满足式（23）。

装配部分同样采用最早完工时间规则确定所有产品的装配顺序。图3所示为一个解码示例。

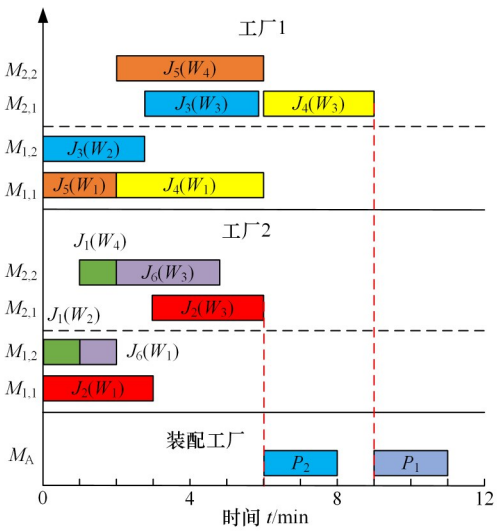



**图3 解码示例**

**Fig.3 Decoding example**

## 2.3 基于知识的初始化策略

基于上述编码，在NEH[21]的框架中使用K1和K2，设计了KD-NEH策略生成初始解[18]。为反映知识K1、K2对解决方案的重要性差异，定义重要性权重：

$$\lambda=N_{f,p}^{a}\Big(\sum_{j,j_1\in I_p}D_{j,j_1}\Big)^{b}\tag{25}$$

式中：$N_{f,p}$为工厂$f$加工产品$p$的工件数；$D_{j,j_1}$为产品$I_p$的工件$j$与工件$j_1$的距离；$a$、$b$分别为K1和K2的权重因子。$N_{f,p}$越小，同一产品在各个工厂的加工越分散；$D_{j,j_1}$越小，同一产品在一个工厂内的分配越集中。

NEH的主要目的是将工件分配到完工时间最短的“位置”，KD-NEH的主要目的是将工件分配到权重与完工时间乘积最小的“位置”。图4展示了基于知识的初始化策略，其中假设$a$、$b$为1，找出权重与完工时间乘积最小的“位置”依次插入工件。

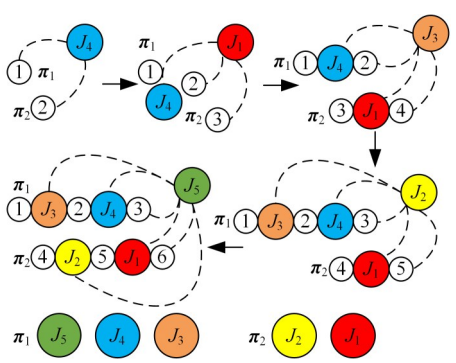



**图4 基于知识的初始化策略示例**

**Fig.4 Example of a knowledge-based initialization strategy**

## 2.4 基于知识的破坏和重构

破坏和重构是传统IG算法的一个重要组成

部分，即从一个完整的解决方案先移除部分工件，再通过NEH重新插入到新的解决方案。为增强KDIG的探索能力，在重新插入工件时采用KB-NEH方法而不是NEH方法[18]。算法1给出了基于知识的破坏和重构伪代码：

算法1：基于知识的破坏和重构

输入：当前解$\pi$，破坏率$d$，权重因子$a$和$b$

输出：新解$\pi_1$

每个产品中的工件按照平均加工时间进行升序排序，获得种子序列$\tau=(\tau_1, \tau_2, \cdots, \tau_N)$，其中，$\tau_i$为在第$i(i=1, 2, \cdots, N)$个位置的工件

$\pi_1=\pi$

$n=$round$(d*N)$

从$\pi_1$中移除$n$个工件，并放入$\tau$

for $i=1$ to $n$ do

　　将$\tau_i$放入$\pi_1$中的所有可能位置，计算每个位置的完工时间和$\lambda$后，将$\tau_i$插入完工时间与$\lambda$乘积最小的位置

end for

Reutrn $\pi_1$

## 2.5　基于Q-learning的变邻域局部搜索

### 2.5.1　基于关键产品的局部搜索

考虑到问题特征，设计了4种基于产品的局部搜索算子(N1～N4)。为更好地表述局部搜索，定义关键产品$P_c$即所有产品中具有最大$T_{TD}$的产品。

N1：随机互换$P_c$中的2个工件，为减少计算，将生成解中的$N_c-1$个解作为最佳解，其中，$N_c$为关键产品$P_c$的工件总数。

N2：从$P_c$中随机选择一个工件，并插入$P_c$中的其他位置，从生成的$N_c-1$个解中选择最佳解。

N3：在$P_c$中随机选择一个工件与$P_c$外的工件交换，从生成的$N-N_c$个解中选择最佳解。

N4：从$P_c$外随机选择一个工件，并插入$P_c$所有可能的位置，从生成的$N-N_c$个解中选择最佳解。

### 2.5.2　Q-Learning策略

采用Q-Learning机制使个体在更新时选择最合适的搜索算子，以克服随机选择的缺点[22]。对于单目标问题，最直观的评价指标为解$\pi$的质量，根据解的质量判断局部搜索前后的个体优劣。

状态集合：KDIG算法中，状态空间的定义是选择动作的关键。不同时间的状态集不同对行动的选择有很大影响。令$s_\upsilon$表示第$\upsilon$种状态，如果个体在执行局部搜索后的解得到改进，则$s_\upsilon=1$，否则$s_\upsilon=0$。

动作集合：令$a_\psi$表示第$\psi$个动作。调度过程中，智能体根据当前状态选择动作。KDIG算法中，动作集为N1～N4。

选择策略：在采用$\varepsilon$贪婪策略选择动作，对应数学表达式为

$$\varepsilon=1-0.9C_{URT}/M_{AXT} \tag{26}$$

式中：$C_{URT}$为当前的CPU运行时间；$M_{AXT}$为最长的CPU运行时间。

奖励函数：在执行动作(N1～N4)后，个体可获得奖励。KDIG算法中，奖励$R$根据最大TTD计算：

$$R=\begin{cases}10 & \pi'>\pi^* \\ 5 & \pi'<\pi^* \\ 0 & \pi'=\pi^*\end{cases} \tag{27}$$

式中：$\pi^*$、$\pi'$分别为搜索前后的解质量。

Q值最后的更新公式为

$$Q_{t+1}(s_t, a_t)=(1-\beta)Q_t(s_t, a_t)+\alpha(R_{t+1}+\Gamma\max(s_{+1}, a)) \tag{28}$$

式中：$\beta$为学习因子，$\beta\in[0,1]$；$R_{t+1}$为$t+1$时刻的奖励；$\Gamma$为衰减因子，$\Gamma\in[0,1]$；$Q_t(s_t,a_t)$表示$t$时刻、状态$s_t$下执行动作$a_t$的所获得的$Q$值。

Q-learning算法有助于个体在进化时自适应选择合适的搜索算子，充分利用各搜索算子的性能，引导种群个体向更好的方向搜索，加快收敛。算法2给出了基于Q-learning的变邻域局部搜索伪代码：

算法2：基于Q-learning的变邻域局部搜索

输入　解$\pi$

输出　最优解$\pi_{best}$

While　不满足终止条件 do

　if $random(0, 1)<\varepsilon$ then

　　从N1～N4中随机选择算子生成新解$\pi'$

　else

　　在N1～N4中随机选择是的Q值最大的算子，生成新解$\pi'$

　end if

　if $T_{TD}(\pi')<T_{TD}(\pi)$ then

　　$\pi=\pi'$

　　if　$T_{TD}(\pi)<T_{TD}(\pi_{best})$ then

　　$\pi_{best}=\pi$

　　end if

　end if

　执行动作后得到相应的奖励$R$

　根据$Q_{t+1}(s_t, a_t)=(1-\beta)Q_t(s_t, a_t)+\alpha(R_{t+1}+\Gamma\max(s_{+1}, a))$更新Q-Table

End while

$\pi=\pi_{best}$

Return $\pi$

## 2.6　接收准则

采用模拟退火接受准则[23]决定是否接受新的解决方案。模拟退火算法中的温度为

$$T=\frac{T_0}{10NN_S}\sum_{i\in I}\sum_{j\in S}\overline{t}_{i,j}^{p} \tag{29}$$

式中：$T_0$为温度系数；$\overline{t}^{\mathrm{P}}_{i,j}$为工件$i$在阶段$j$的平均处理时间。

提出策略：如果0～1之间的随机数$r$满足$r\leqslant\exp(T_{\mathrm{TD}}(\boldsymbol{\pi})-T_{\mathrm{TD}}(\boldsymbol{\pi}'))/T$，则将较差解$\boldsymbol{\pi}'$代替当前解$\boldsymbol{\pi}$(对应概率为$r$)，从而帮助KDIG算法跳出局部最优。

## 3 实验比较与分析

### 3.1 实验设置

为评估KDIG算法解决DAHFSSP-DRC的性能，使用实例进行测试，其中，$N\in\{20,40,60\}$，$N_{\mathrm{P}}\in\{2,4,6\}$，$N_{\mathrm{S}}\in\{2,4,6\}$，$N_{\mathrm{F}}\in\{2,4,6\}$。4个参数不同组合可生成81个大型实例。实例的加工时间是区间[1,200]内的随机数。每个产品中的组成工件随机选择。每个阶段(生产中的一个特定工序或加工步骤)的工人固定为2人。每个阶段使用的机器编号均从2、3中随机抽取。产品的装配时间$T_{\mathrm{A}}=\overline{T}_{\mathrm{m}}R_1$，其中，$\overline{T}_{\mathrm{m}}$为加工部分的平均加工时间，$R_1$为[0.1,0.2]内随机数。每个产品的交货日期$T_{\mathrm{D}}=\overline{T}_{\mathrm{m}}R_2/N_{\mathrm{F}}$，其中，$R_2$为[0.5,0.6]内随机数。算例已上传，通过链接https://github.com/CUGYU/DAHFSSP DRCdata可获取。

所有实例的每个算法都重复运行20次，停止准则为CPU最长运行时间$M_{\mathrm{AXT}}=0.1NN_{\mathrm{S}}N_{\mathrm{F}}N_{\mathrm{P}}$。评价指标为平均相对偏差

$$A_{\mathrm{RPD}}=\frac{1}{R}\sum_{i=1}^{R}\frac{C_i-C_{\mathrm{best}}}{C_{\mathrm{best}}} \tag{30}$$

式中：$C_i$为KDIG算法在第$i$次测试中获得的目标值；$C_{\mathrm{best}}$为测试获得的最小目标值。

### 3.2 参数校准

采用文献[24]的实验设计方法，详细分析参数对所提算法的影响，其中，学习因子$\beta\in\{0.1,0.2,0.3,0.4\}$，衰减因子$\Gamma\in\{0.5,0.6,0.7,0.8\}$，温度系数$T_0\in\{0.4,0.8,1.2,1.6\}$，权重因子$a$，$b\in\{0.5,1.5,2.0\}$。上述参数的组合总数为576，随机选择60个组合的算例校准参数。每个实例在1个参数组合下运行20次，比较20次结果的$A_{\mathrm{RPD}}$。实验结果采用多元方差分析(analysis of variance，ANOVA)进行分析，验证参数之间的交互作用是否具有统计学意义。表2所示为多元方差的分析结果。

如果$p$-value小于阈值0.05，则对应的2个变量存在显著差异。如表2所示，学习因子$\beta$对应的$F$-ratio在所有测试变量中最大，对算法有明显的影响。KDIG算法参数主效应如图5所示。结合表2和图5得到KDIG算法的参数取值：$\beta=0.3$，$\Gamma=0.8$，$T_0=1.2$，$a=1.5$，$b=2.0$。

**表2 多元方差分析结果**

**Tab.2 Results of multivariate ANOVA**

| 来源 | 平方和 | 自由度 | 均方 | $F$-ratio | $p$-value |
|---|---|---|---|---|---|
| $\beta$ | 456.8 | 3 | 86.582 | 891.76 | **0.0000** |
| $\Gamma$ | 352.3 | 3 | 69.485 | 423.26 | **0.0000** |
| $T_0$ | 6.9 | 3 | 3.852 | 65.42 | **0.0000** |
| $a$ | 3.2 | 2 | 1.262 | 9.59 | **0.0000** |
| $b$ | 2.5 | 2 | 1.258 | 7.66 | **0.0000** |
| $\beta\times\Gamma$ | 39.8 | 9 | 3.256 | 1.78 | 0.6629 |
| $\beta\times T_0$ | 0.6 | 9 | 0.266 | 1.56 | 0.8413 |
| $\Gamma\times a$ | 2.5 | 6 | 0.521 | 1.44 | 0.5130 |
| $\Gamma\times b$ | 6.4 | 6 | 0.636 | 4.56 | 0.4790 |
| $\Gamma\times T_0$ | 3.5 | 9 | 0.466 | 3.85 | 0.5364 |
| $\beta\times a$ | 1.1 | 6 | 0.226 | 2.65 | 0.6449 |
| $\beta\times b$ | 0.9 | 6 | 0.255 | 1.25 | 0.7418 |
| $T_0\times a$ | 0.3 | 6 | 0.236 | 0.45 | 0.8234 |
| $T_0\times b$ | 0.2 | 6 | 0.126 | 1.34 | 0.6950 |
| $a\times b$ | 0.1 | 4 | 0.236 | 0.45 | 0.8074 |
| 残差 | 6964.0 | 76728 | | | |
| 总计 | 7603.2 | 76800 | | | |

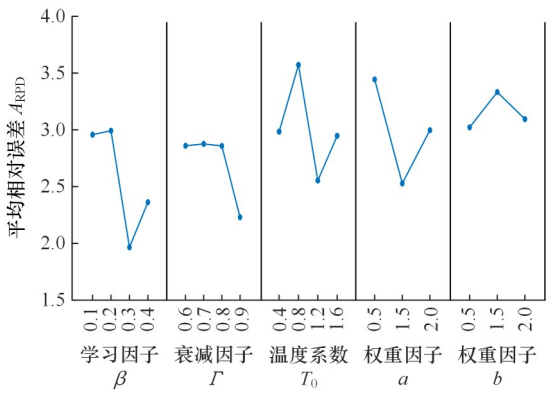



**图5 KDIG算法的参数主效应图**

**Fig.5 Parameter main effect diagram of KDIG algorithm**

### 3.3 组成成分分析

为证明KDIG算法中各个策略的有效性，设计变体KDIG1～KDIG3与KDIG进行对比实验。其中，KDIG1采用NEH策略初始种群，其他策略保持不变；KDIG2采用NEH插入方法替换基于知识的破坏和构建方式，其他策略保持不变；KDIG3去除Q-learning机制，其他策略保持不变。

表3给出了KDIG算法及其变体在81个大规模实例的$A_{\mathrm{RPD}}$(KDIG算法及其变体对每个实例各自独立执行20次的结果)，最优结果以粗体显示。为分析KDIG算法与其变体是否具有显著差异，进行了Wilcoxon检验，结果见表4，其中，$R^+$表示结果可取，$R^-$则相反。图6所示为KDIG算法及其变体算法的结果区间图。

分析表3和图6可知KDIG算法优于变体。

表3　KDIG算法及变体的$A_{RPD}$

Tab.3　$A_{RPD}$ for KDIG algorithm and its variants

| 参数 | 规模 | KDIG | KDIG1 | KDIG2 | KDIG3 |
|---|---|---|---|---|---|
| $N_F$ | 2 | **1.15** | 17.52 | 7.63 | 23.64 |
| | 4 | **2.39** | 10.96 | 5.95 | 18.89 |
| | 6 | **4.05** | 9.85 | 4.25 | 15.89 |
| $N$ | 20 | **3.54** | 4.05 | 3.08 | 14.68 |
| | 40 | **0.96** | 11.48 | 3.09 | 20.65 |
| | 60 | **0.25** | 22.85 | 5.67 | 21.66 |
| $N_S$ | 2 | **2.90** | 12.58 | 8.94 | 20.69 |
| | 4 | **2.39** | 13.57 | 6.84 | 16.36 |
| | 6 | **2.39** | 11.78 | 6.04 | 14.65 |
| $N_P$ | 2 | **2.85** | 11.74 | 5.09 | 19.67 |
| | 4 | **1.68** | 13.85 | 6.98 | 18.65 |
| | 6 | **3.18** | 12.45 | 6.28 | 18.55 |

表4　KDIG算法及变体的Wilcoxon结果

Tab.4　Wilcoxon results of KDIG algorithm and its variants

| | $R^+$ | $R^-$ | $p$-value |
|---|---|---|---|
| KDIG vs KDIG1 | 2891.0 | 430.0 | **0.00000** |
| KDIG vs KDIG2 | 2512.0 | 1124.0 | **0.00006** |
| KDIG vs KDIG3 | 3365.0 | 44.0 | **0.00000** |

Wilcoxon检验得到的$p$-value均小于0.05，证实KDIG算法与其变体存在显著差异。对比KDIG1和KDIG可知，在传统NEH策略中添加问题知识可提高算法的性能。KDIG3在所有变体中表现最差，说明基于Q-learning的变邻域局部搜索对KDIG算法性能的影响最大。综上可得，所提策略对KDIG算法解决DAHFSSP-DRC都有效，但不同策略对KDIG算法的性能有不同程度的影响。

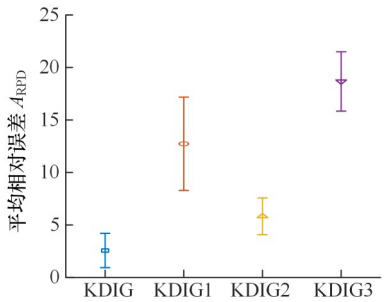



图6　KDIG算法及其变体结果区间图

Fig.6　Result interval graph of KDIG algorithm and its variants

### 3.4　算法比较

为证明KDIG算法的有效性，将其与算法KBIG[18]、HDIWO[25]、IG_VNS[26]、MNIG[27]和WWO[28]进行比较。分别采用6种算法对每个实例各自独立执行20次，获取对应的$A_{RPD}$。由表5可知KDIG算法在大多数组上的结果明显优于其他比较算法。

为分析KDIG算法与对比算法是否具有显著差异，进行Wilcoxon检验，结果见表6，所有的$p$-value均小于0.05，这表明KDIG算法与其他算法差异显著。如图7所示，KDIG算法的$A_{RPD}$均最小，再次验证KDIG算法的最优性。

表5　KDIG算法及对比算法的$A_{RPD}$

Tab.5　$A_{RPD}$ of KDIG algorithm and comparison algorithms

| 参数 | 规模 | KDIG | HDIWO | IG_VNS | MNIG | WWO | KBIG |
|---|---|---|---|---|---|---|---|
| $N_F$ | 2 | **1.15** | 4.41 | 15.10 | 25.72 | 16.47 | 7.15 |
| | 4 | **2.39** | 5.64 | 6.60 | 26.53 | 9.05 | 4.04 |
| | 6 | **4.05** | 7.37 | 4.54 | 20.61 | 14.66 | 4.86 |
| $N$ | 20 | **3.54** | 10.21 | 2.95 | 4.81 | 5.20 | 7.02 |
| | 40 | **0.96** | 3.35 | 7.43 | 23.05 | 12.10 | 1.63 |
| | 60 | **0.25** | 1.67 | 15.33 | 24.03 | 22.8 | 1.19 |
| $N_S$ | 2 | **2.90** | 7.60 | 9.83 | 29.59 | 10.11 | 4.25 |
| | 4 | **2.39** | 3.94 | 8.76 | 30.02 | 10.18 | 2.80 |
| | 6 | **2.39** | 3.87 | 7.12 | 30.58 | 19.57 | 2.77 |
| $N_P$ | 2 | **2.85** | 6.60 | 6.39 | 27.58 | 12.85 | 3.00 |
| | 4 | **1.68** | 3.48 | 9.33 | 33.09 | 16.89 | 2.84 |
| | 6 | **3.18** | 5.20 | 9.99 | 30.28 | 11.03 | 4.00 |

表6　KDIG算法及对比算法Wilcoxon结果

Tab.6　Wilcoxon results of KDIG algorithm and comparison algorithms

| | $R^+$ | $R^-$ | $p$-value |
|---|---|---|---|
| KDIG vs HDIWO | 2480.0 | 841.0 | **0.00000** |
| KDIG vs IG_VNS | 2471.0 | 850.0 | **0.00000** |
| KDIG vs MNIG | 2887.0 | 434.0 | **0.00000** |
| KDIG vs WWO | 2754.0 | 567.0 | **0.00000** |
| KDIG vs KBIG | 2356.0 | 858.0 | **0.00007** |

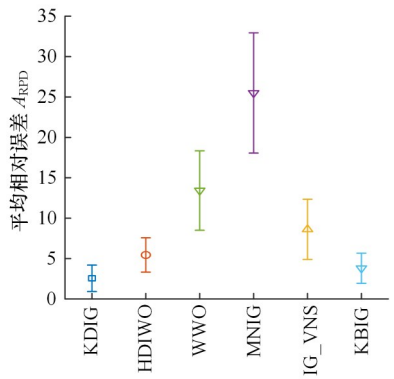



图7　KDIG算法及其对比算法结果区间图

Fig.7　The results of the KDIG algorithm and its comparative algorithms in the form of an interval graph

为进一步验证KDIG算法的收敛性，给出两个不同规模算例的收敛曲线。图8a为算例$N$=20，$N_S$=$N_F$=$N_F$=2的收敛曲线，所有算法初始解的差距不大，但KDIG算法的下降趋势最为明显，说明其优势较大。图8b的结果与图8a类似，KDIG算法在收敛性方面也是最优。由此可得，KDIG算法优于算法KBIG、HDIWO、IG_VNS、MNIG和WWO。

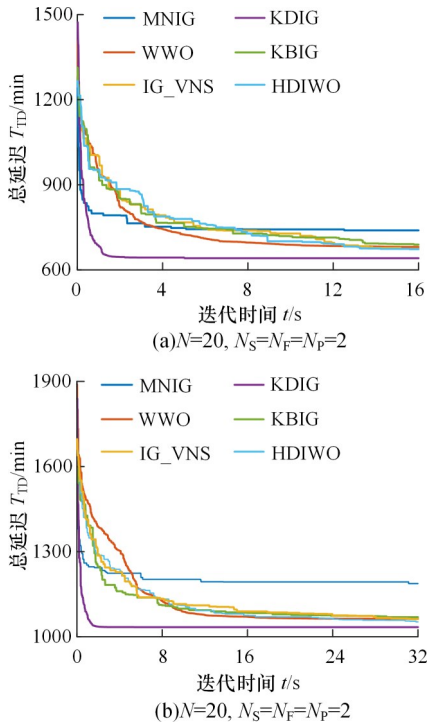



**图8 KDIG算法以及对比算法的收敛曲线图**

**Fig.8 Curves of KDIG algorithm and comparison algorithms**

## 4 结论

针对以最小化总延迟为目标的DAHFSSP-DRC求解，设计了一种KDIG算法。初始化阶段，采用KD-NEH启发式初始化策略生成初始解决方案；破坏和重构阶段，设计基于知识的破坏和构造策略，以提高勘探能力；局部搜索阶段，提出基于Q-learning的变邻域局部搜索，以提高开发能力。在81个大型实例上进行了KDIG算法与其他5种算法的比较实验。实验结果表明，KDIG算法优于其他算法。

**参考文献：**

[1] SHAO Zhongshi, SHAO Weishi, PI Dechang. LS-HH: a Learning-based Selection Hyper-heuristic for Distributed Heterogeneous Hybrid Blocking Flow-shop Scheduling[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2023, 7(1): 111-127.

[2] ZHANG Wenqiang, LI Chen, GEN M, et al. A Multiobjective Memetic Algorithm with Particle Swarm Optimization and Q-learning-based Local Search for Energy-efficient Distributed Heterogeneous Hybrid Flow-shop Scheduling Problem [J]. Expert Systems with Applications, 2024, 237: 121570.

[3] DI Yuanzhu, DENG Libao, ZHANG Lili. A Collaborative-learning Multi-agent Reinforcement Learning Method for Distributed Hybrid Flow Shop Scheduling Problem [J]. Swarm and Evolutionary Computation, 2024, 91: 101764.

[4] LIU Feige, LI Guiling, LU Chao, et al. A Tri-individual Iterated Greedy Algorithm for the Distributed Hybrid Flow Shop with Blocking [J]. Expert Systems with Applications, 2024, 237: 121667.

[5] CUI Hanghao, LI Xinyu, GAO Liang, et al. Multi-population Genetic Algorithm with Greedy Job Insertion Inter-factory Neighbourhoods for Multi-objective Distributed Hybrid Flow-shop Scheduling with Unrelated-parallel Machines Considering Tardiness[J]. International Journal of Production Research, 2024, 62(12): 4427-4445.

[6] SHAO Weishi, SHAO Zhongshi, PI Dechang. A Network Memetic Algorithm for Energy and Labor-aware Distributed Heterogeneous Hybrid Flow Shop Scheduling Problem[J]. Swarm and Evolutionary Computation, 2022, 75: 101190.

[7] 魏光艳，叶春明. 分布式多柔性装配作业车间调度问题研究[J]. 中国机械工程，2023，34(20)：2442-2455.
WEI Guangyan, YE Chunming. Research on Distributed and Multi-flexible Assembly Job-shop Scheduling Problems[J]. China Mechanical Engineering, 2023, 34(20): 2442-2455.

[8] 曹阳华，孔繁森. 基于装配线平衡的U形装配线生产效率研究[J]. 中国机械工程，2015，26(14)：1908-1915.
CAO Yanghua, KONG Fansen. Study on Production Efficiency of U-shaped Assembly Line Based on Assembly Line Balancing[J]. China Mechanical Engineering, 2015, 26(14): 1908-1915.

[9] 魏书鹏，唐红涛，李西兴，等. 考虑双资源约束的柔性机械加工车间逆调度问题研究[J]. 中国机械工程，2024，35(3)：457-471.
WEI Shupeng, TANG Hongtao, LI Xixing, et al. Dual-resource Constrained Flexible Machining Workshop Inverse Scheduling Problem [J]. China Mechanical Engineering, 2024, 35(3): 457-471.

[10] 梁向檩，宋豫川，雷琦，等. 考虑工人数量配置优化的柔性作业车间调度问题研究[J]. 中国机械工程，2023，34(17)：2065-2076.
LIANG Xianglin, SONG Yuchuan, LEI Qi, et al. Research on Flexible Job-shop Scheduling Problems Considering Optimization of Worker Number Allocation[J]. China Mechanical Engineering, 2023, 34(17): 2065-2076.

[11] 张维存，顾洪羽. 一人多机模式下考虑相似学习效应的车间调度问题[J]. 中国机械工程，2023，34(14)：1701-1709.
ZHANG Weicun, GU Hongyu. Job-shop Scheduling Problems Considering Similar Learning Effect in One-worker and Multiple-machine Partterns[J]. China Mechanical Engineering, 2023, 34(14): 1701-1709.

[12] RUIZ R, STÜTZLE T. A Simple and Effective Iterated Greedy Algorithm for the Permutation Flowshop Scheduling Problem [J]. European Journal of Operational Research, 2007, 177(3): 2033-2049.

[13] JING Xuelei, PAN Quanke, GAO Liang, et al. An Effective Iterated Greedy Algorithm for a Robust Distributed Permutation Flowshop Problem with Carryover Sequence-dependent Setup Time [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(9): 5783-5794.

[14] ZOU Wenqiang, PAN Quanke, MENG Leilei, et al. An Effective Self-adaptive Iterated Greedy Algorithm for a Multi-AGVs Scheduling Problem with Charging and Maintenance [J]. Expert Systems with Applications, 2023, 216: 119512.

[15] QIN Haoxiang, HAN Yuyan, ZHANG Biao, et al. An Improved Iterated Greedy Algorithm for the Energy-efficient Blocking Hybrid Flow Shop Scheduling Problem[J]. Swarm and Evolutionary Computation, 2022, 69: 100992.

[16] ZHAO Fuqing, XU Zesong, WANG Ling, et al. A

Population-based Iterated Greedy Algorithm for Distributed Assembly No-wait Flow-shop Scheduling Problem [J]. IEEE Transactions on Industrial Informatics, 2023, 19(5): 6692-6705.
[17] KARIMI-MAMAGHAN M, MOHAMMADI M, PASDELOUP B, et al. Learning to Select Operators in Metaheuristics: an Integration of Q-learning into the Iterated Greedy Algorithm for the Permutation Flowshop Scheduling Problem [J]. European Journal of Operational Research, 2023, 304(3): 1296-1330.
[18] YU Fei, LU Chao, ZHOU Jiajun, et al. Mathematical Model and Knowledge-based Iterated Greedy Algorithm for Distributed Assembly Hybrid Flow Shop Scheduling Problem with Dual-resource Constraints [J]. Expert Systems with Applications, 2024, 239: 122434.
[19] YANG Yahong, LI Xun. A Knowledge-driven Constructive Heuristic Algorithm for the Distributed Assembly Blocking Flow Shop Scheduling Problem [J]. Expert Systems with Applications, 2022, 202: 117269.
[20] RUIZ R, MAROTO C. A Genetic Algorithm for Hybrid Flowshops with Sequence Dependent Setup Times and Machine Eligibility [J]. European Journal of Operational Research, 2006, 169(3): 781-800.
[21] NAWAZ M, ENSCORE E E, HAM I. A Heuristic Algorithm for the M-machine, N-job Flow-shop Sequencing Problem [J]. Omega, 1983, 11(1): 91-95.
[22] ZHAO Fuqing, HU Xiaotong, WANG Ling, et al. A Reinforcement Learning Brain Storm Optimization Algorithm with Learning Mechanism [J]. Knowledge Based Systems, 2022, 235: 107645.
[23] KOU Lei, WANG Yukuan, ZHANG Fangfang, et al. A Chaotic Simulated Annealing Genetic Algorithm with Asymmetric Time for Offshore Wind Farm Inspection Path Planning [J]. International Journal of Bio-Inspired Computation, 2025, 25(2): 69-78.
[24] SMITH J R, LARSON C. Statistical Approaches in Surface Finishing. Part 3. Design-of-experiments [J]. Transactions of the IMF, 2019, 97(6): 289-294.
[25] SANG Hongyan, PAN Quanke, LI Junqing, et al. Effective Invasive Weed Optimization Algorithms for Distributed Assembly Permutation Flowshop Problem with Total Flowtime Criterion [J]. Swarm and Evolutionary Computation, 2019, 44: 64-73.
[26] SHAO Weishi, PI Dechang, SHAO Zhongshi. Optimization of Makespan for the Distributed No-wait Flow Shop Scheduling Problem with Iterated Greedy Algorithms [J]. Knowledge-Based Systems, 2017, 137: 163-181.
[27] SHAO Weishi, SHAO Zhongshi, PI Dechang. Modeling and Multi-neighborhood Iterated Greedy Algorithm for Distributed Hybrid Flow Shop Scheduling Problem [J]. Knowledge-Based Systems, 2020, 194: 105527.
[28] ZHAO Fuqing, SHAO Dongqu, WANG Ling, et al. An Effective Water Wave Optimization Algorithm with Problem-specific Knowledge for the Distributed Assembly Blocking Flow-shop Scheduling Problem [J]. Knowledge-Based Systems, 2022, 243: 108471.

（编辑　张　洋）

**作者简介：赵　才**，男，1996年生，博士研究生。研究方向为生产调度、智能算法。发表论文10余篇。E-mail：zhaocai1996@163.com。**吴亮红***（通信作者），男，1977年生，教授，博士研究生导师。研究方向为智能优化与调度、多目标优化。发表论文100余篇。E-mail：lhwu@hnust.edu.cn.

**本文引用格式：**

赵才，吴亮红．具有双资源约束的分布式装配混合流水车间调度[J]．中国机械工程，2026，37(7)：1725-1733.

ZHAO Cai, WU Lianghong. Distributed Assembly Hybrid Flow Shop Scheduling with Dual Resource Constraints [J]. China Mechanical Engineering, 2026, 37(7): 1725-1733.

（上接第1724页）

WEI Jinwen, QIN Hequn, et al. Recognizing Data Fitting Based on the Robustness of Generalized Cross Product [J]. Journal of Mechanical Engineering, 2011, 47(14): 7-12.
[21] 李振，赵欢，王辉，等．机器人磨抛加工接触稳态自适应力跟踪研究[J]．机械工程学报，2022，58(9)：200-209.
LI Zhen, ZHAO Huan, WANG Hui, et al. Research on Contact Steady-state Adaptive Force Tracking of Robot Grinding and Polishing [J]. Journal of Mechanical Engineering, 2022, 58(9): 200-209.
[22] 周华伟，王成明，孙大万，等．基于简化有限集模型预测电流控制的五相PMSM统一容错控制[J]．中国电机工程学报，2024，44(1)：269-279.
ZHOU Huawei, WANG Chengming, SUN Dawan, et al. A Unified Fault-tolerant Control of Five-phase PMSM Based on Simplified Finite Control Set Model Predictive Current Control [J]. Proceedings of the CSEE, 2024, 44(1): 269-279.
[23] 宋筱轩，林振伟，赵丽娜，等．具有固定时间规定性能控制的无人船自适应状态反馈控制[J]．自动化与仪器仪表，2026(1)：115-120.
SONG Xiaoxuan, LIN Zhenwei, ZHAO Lina, et al. Adaptive State Feedback Control for Unmanned Surface Vessels with Fixed-time Prescribed Performance [J]. Automation & Instrumentation, 2026(1): 115-120.
[24] 潘柏松，俞铭杰，项涌涌，等．考虑刀具磨损的铣削加工精度可靠性分析及工艺优化设计[J]．计算机集成制造系统，2020，26(11)：2982-2991.
PAN Bai|Bo) Song), YU Mingjie, XIANG Yongyong, et al. Accuracy Reliability Analysis and Process Optimization Design of Milling Processing Considering Tool Wear [J]. Computer Integrated Manufacturing Systems, 2020, 26(11): 2982-2991.

（编辑　张　洋）

**作者简介：韦进文***（通信作者），男，1976年生，教授。研究方向为机械电子工程。发表论文30篇。E-mail：my595@sina.com.cn。

**本文引用格式：**

韦进文，王肃鹏，李东旭，等．机床位置伺服系统的正交反馈-前馈补偿方法[J]．中国机械工程，2026，37(7)：1724-1732.

WEI Jinwen, WANG Supeng, LI Dongxu, et al. Orthogonal Feedback-Feedforward Compensation Method for Machine Tool Position Servo Systems [J]. China Mechanical Engineering, 2026, 37(7): 1724-1732.