

改进果蝇算法求解混合流水车间调度问题

周永强, 王翠雨[†], 李颖俐, 李新宇

(华中科技大学 数字制造装备与技术国家重点实验室, 湖北 武汉 430074)

摘要: 针对混合流水车间调度问题(HFSP), 本文提出了一种新的基于果蝇算法和变邻域搜索的混合优化方法. 首先, 将关键块内的工序与同阶段其他机器上的工序进行交换, 提出了一种基于关键路径的HFSP新邻域结构. 其次, 针对HFSP的阶段式解码特性, 提出了一种邻域解的快速评估方法, 并验证了快速评估方法的高效性. 然后, 基于提出的新邻域结构, 并将N7和K-insertion邻域结构引入HFSP, 设计了基于上述3种邻域结构的变邻域搜索方法, 以此为基础提出了一种针对HFSP的混合优化方法. 最后, 通过对Carlier和Liao等经典测试集进行测试, 验证了所提新邻域结构的可行性和有效性, 并将该方法与其他文献的方法进行了对比, 验证了所提方法的优越性.

关键词: 混合流水车间调度; 关键路径; 快速评估; 果蝇算法

引用格式: 周永强, 王翠雨, 李颖俐, 等. 改进果蝇算法求解混合流水车间调度问题. 控制理论与应用, 2023, 40(4): 597 – 606

DOI: 10.7641/CTA.2022.10962

Improved fruit fly optimization algorithm for solving the hybrid flow shop scheduling problem

ZHOU Yong-qiang, WANG Cui-yu[†], LI Ying-li, LI Xin-yu

(State Key Lab of Digital Manufacturing Equipment & Technology,
Huazhong University of Science & Technology, Wuhan Hubei 430074, China)

Abstract: Aiming at the hybrid flow-shop scheduling problem (HFSP), this paper proposes a new hybrid optimization method based on the fruit fly algorithm and variable neighbourhood search (VNS). Firstly, a new neighbourhood structure of HFSP based on the critical path is proposed by the operation of exchanging within the critical block. Secondly, a rapid evaluation method of neighbourhood solutions is proposed for the stage-based decoding method of HFSP, and the efficiency of this evaluation method is verified. Then, based on the proposed new neighbourhood structure and introducing the N7 and K-insertion neighbourhood structures into HFSP, a variable neighbourhood search method based on the above three neighbourhood structures is designed as a basis for proposing a hybrid optimisation method for HFSP. Finally, the feasibility and effectiveness of the proposed new neighbourhood structure is verified by testing it on classical benchmark such as Carlier and Liao, and the method is compared with other methods in the literature to verify the superiority of the proposed method.

Key words: hybrid flow shop scheduling; critical path; fast evaluation; fruit fly algorithm

Citation: ZHOU Yongqiang, WANG Cuiyu, LI Yingli, et al. Improved fruit fly optimization algorithm for solving the hybrid flow shop scheduling problem. *Control Theory & Applications*, 2023, 40(4): 597 – 606

1 引言

混合流水车间是由多道工序且每道工序具有一台或多台并行机器构成的流水式生产车间^[1], 其优势在于有效地平衡机器载荷、提高产能, 在造纸、化工、制药、纺织等^[2-3]领域具有广泛地应用. 因此, 对混合流水车间调度问题(hybrid flow-shop scheduling problem, HFSP)进行研究具有重要意义. 对比流水车间调度问题, HFSP增加了工序的可选机器柔性, 导致求解难度

增大, 是NP-hard问题. 在求解HFSP问题中, 分布式估计算法(estimation of distribution algorithm, EDA)、差分进化(differential evolution, DE)算法、人工蜂群算法(artificial bee colony algorithm, ABC)、灰狼算法(grey wolf optimizer, GWO)、鲸鱼群算法 (whale swarm algorithm, WSA)以及蛙跳算法(shuffled frog-leaping algorithm, SFLA)等^[4-7]智能优化方法得到广泛应用. 这类算法搜索速度快, 能在较短时间得出较大规模问

收稿日期: 2021-10-11; 录用日期: 2022-03-01.

[†]通信作者. E-mail: 76325434@qq.com.

本文责任编辑: 王凌.

国家重点研发计划项目(2019YFB1704600)资助.

Supported by the National Key Research and Development Program (2019YFB1704600).

题的较优解. Wang等^[7]针对具有相同并行机的HFSP, 提出了改进EDA. 张源等^[8]结合反向学习策略和自适应交叉变异因子, 提出了改进DE. Pan等^[9]结合正向和逆向解码策略, 提出了离散的ABC. Li等^[10]针对带序列相关准备时间的分布式HFSP, 结合关键车间的邻域结构, 提出了改进的离散ABC. 随后, Li等^[11]将问题拓展为异构的车间结构, 设计了改进的ABC. Lu等^[12]考虑噪声污染的多目标HFSP, 提出了改进的GWO. Zhang等^[13]针对具有有限缓冲区和两条工艺路线的HFSP, 提出了离散的WSA. 董君等^[14]针对加工时间不确定的可重入HFSP, 提出了改进WSA以实现制造与维护的联合优化. 雷德明等^[15]针对低碳HFSP, 提出了新型的SFLA, 优化了总能耗和总延迟时间. 同时, 针对分布式两阶段HFSP, 提出了改进的SF-LA^[16]. 果蝇算法^[17] (fruit fly optimization algorithm, FOA)是基于果蝇觅食行为所提出的群智能优化算法, 其具有操作简单、控制参数少、易于实现等特点. 2015年, 王凌与郑晓龙^[18]提出了3种改进方向: 维持种群多样性、知识驱动和协同机制^[19]. 此外, Wang等^[20]还提出了一种变异机制, 使算法跳出局部最优. Zheng等^[21]针对考虑资源约束的两阶段不相关并行机调度问题, 提出了自适应的FOA. Shao等^[22]针对带阻塞的分布式流水车间, 提出了离散的FOA. 王凌等^[23]针对不确定分布式置换流水线调度问题, 提出了混合离散FOA. Wang等^[24]针对多技能资源约束的项目调度问题, 提出了知识引导的多目标FOA. Duan等^[25]将多种群和竞争机制引入FOA, 以优化HFSP. 综上所述, FOA在调度问题上具有较大优势. 因此, 本文以FOA算法为框架, 提出了求解HFSP的混合优化方法, 设计了基于关键路径的HFSP新邻域结构, 并结合VNS(variable neighborhood search)^[26]改进了FOA. 为提高FOA的计算速度, 设计了一种快速评估方法, 对邻域解进行了快速而准确的评估, 目前还未有研究针对HFSP设计邻域解的快速评估方法. 最后, 通过对Carlier和Liao两类经典测试集进行测试, 验证了所提新邻域结构的可行性和有效性, 并将该方法与其他文献的方法进行了对比, 验证了所提方法的优越性.

2 混合流水车间调度问题概述

HFSP描述如下: n 个工件在 m 台机器上加工 S 个阶段, 其中每个阶段至少存在一台机器, 且至少有一个阶段存在两台或两台以上的并行机器. 各工件需要在每个阶段上完成一道工序, 且工件可以选择该阶段的任意一台机器进行加工. 需要确定所有加工阶段的工件排序, 以及工件的机器分配, 以实现目标函数的最小化. 通常有以下假设^[2]:

- 1) 工件在同一时刻只能在一台机器上加工;
- 2) 一台机器同一时刻只能加工一个工件;

- 3) 当前工序加工完成后才能进行下一道工序;
- 4) 工件可选择当前阶段的任一机器进行加工;
- 5) 工件一旦开始加工便不可中断.

3 基于关键路径的HFSP新邻域结构

邻域移动是对给定解进行扰动, 从而获得新解的更新机制^[27]. 若无邻域结构的约束, 不仅会导致无规律的邻域移动, 也将耗费时间来评估移动后的适应度值. 因此, 在进行邻域搜索时, 需要根据问题特征设计邻域结构, 在保证解质量的前提下, 尽量减少不必要的移动. 其中: 当以最小化最大完工时间为目标时, 移动关键路径(从加工开始到加工结束的最长加工路径)上的工序是优化 $C_{\max}$ 的有效手段. 因此, 在选择可移动的工序时, 本文针对关键路径上的工序进行操作, 可提高搜索效率. 本文引入了适用于柔性作业车间调度问题(flexible job shop scheduling problem, FJSP)的邻域结构, 并根据已有的FJSP邻域结构和HFSP的关键路径特性, 本文提出了一种新的邻域结构.

3.1 同阶段机器内的工序插入的邻域结构

针对同阶段机器内的工序插入, Zhang^[27]提出了N7邻域结构, 该方法将关键块内的工序移动到块首之前或块尾之后, 或将关键块的块首工序或块尾工序移动至关键块内部, 以提高解的质量. 由于HFSP解码特性, 将该邻域结构引入HFSP, 不会产生不可行解. 因此, 本文将N7邻域结构用于算法的局部搜索, 以提高解的质量.

3.2 同阶段机器间的工序插入的邻域结构

针对FJSP的同阶段机器间的工序插入, Mastrolilli和Gambardella^[28]提出K-insertion操作, 使得关键块上的工件以某一特定的规律进行移动不会使解变差. 由于该邻域结构在HFSP中不会产生不可行邻域解, 本文将上述邻域结构引入至HFSP中.

3.3 同阶段机器间的工序交换的新邻域结构

目前, 针对邻域结构的研究, 多数学者仅进行一步预测(如插入操作), 且只针对当前个体邻域解中最优的个体进行进一步搜索, 以获取更优解. 若仅针对最优邻域解进行更深层次的搜索, 则会因解码规则的固定性, 导致搜索限制过大, 从而限制了一步预测中较差解生成更优解的能力. 因此, 本文将一步预测扩大至两步预测, 以接受在第1步搜索时变差而在第2步搜索时变好的邻域解.

本文基于两步预测构建了新的邻域结构: 同阶段机器间的工序交换, 如图1所示. 机器 m_1 上的关键块如图所示, 将工序 O_{m_14} 与 O_{m_24} 进行机器间的工序交换, 该交换过程首先将工序 O_{m_14} 移动至 O_{m_25} 之前, 然后将工序 O_{m_24} 移动至 O_{m_15} 之前, 以此进行两步插入操作. 由于HFSP同阶段上各工件可选择机器相同, 在进行机器间的工序互换时, 不会产生不可行解, 保证

了移动的可行性.

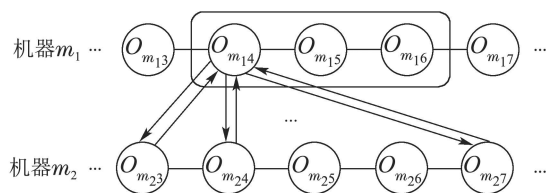


图1 同阶段机器间的工序交换

Fig. 1 Swap operate between the same stage

表1给出了 5×3 算例的加工信息. 针对该算例的一个可行解, 通过交换同阶段机器间的工序, 优化了解的完工时间, 验证了新邻域结构的有效性.

表1 5×3 算例

Table 1 5×3 instance

阶段	机器数	加工时间				
		工件1	工件2	工件3	工件4	工件5
1	3	2	4	5	1	6
2	2	7	8	3	6	3
3	3	5	9	4	7	2

表1中, 根据3, 5, 2, 4, 1的工序编码, 利用先到先加工的解码方式进行解码, 所得的甘特图如图2所示. 将机器M4上的工序1与机器M5上的工序5交换, 所得甘特图如图3所示. 对比两图, 目标函数值由26优化到了24.

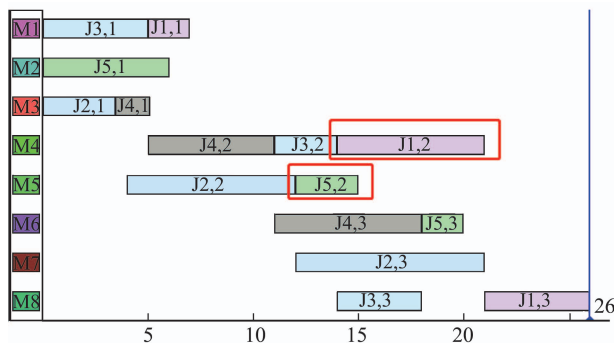


图2 同阶段机器间的工序交换前

Fig. 2 Before swap operate between the same stage

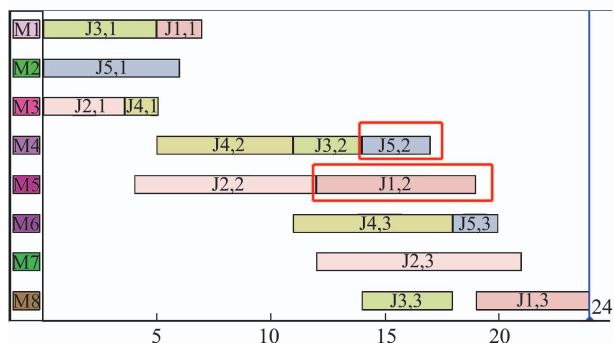


图3 同阶段机器间的工序交换后

Fig. 3 After swap operate between the same stage

4 HFSP邻域解的快速评估方法

判断邻域解的优劣, 需要对其进行重新解码以获取适应度值. 但一次邻域移动只调整了小部分机器上的局部工序, 该机器上的其余工序仍保持不变, 多次解码会耗费大量运算时间. 因此, 本文结合HFSP的特性, 设计了一种新的快速评估方法.

HFSP解码是通过上阶段的完工时间确定当前阶段工件的加工顺序. 基于这种阶段式的解码方式, 本文将邻域移动前后的调度方案分为3个部分: 若当前操作是针对阶段 a , 则 $(0, a)$ 阶段为第1部分, a 阶段为第2部分, (a, S) 阶段为第3部分. 设邻域操作前各工序的头时间为 s 、尾时间为 t , 邻域操作后各工序的头时间为 s^- 、尾时间为 t^- , 同机器上前序工序为MP, 同机器上后序工序为MS, P_{ij} 为工件 j 在阶段 i 的加工时间. 在邻域操作前后, 3部分中仅第2部分会调整部分机器上的工件序列, 第1和第3部分都将保持邻域移动前的工件序列. 因此, 第1部分总满足 $s = s^-$, 第3部分总满足 $t = t^-$.

在开始加工到加工结束的过程中, 总会存在一条关键路径是经过工序 O_{aj} , 那么

$$\text{makespan} = \max \{j \in J | s_{O_{aj}}^- + t_{O_{aj}}^- + P_{O_{aj}}^-\}, \quad (1)$$

其中:

$$s_{O_{aj}}^- = \max \{s_{O_{aj-1}}^- + P_{O_{aj-1}}^-, s_{O_{aj,MP}}^- + P_{O_{aj,MP}}^-\} = \max \{s_{O_{aj-1}} + P_{O_{aj-1}}^-, s_{O_{aj,MP}} + P_{O_{aj,MP}}^-\}, \quad (2)$$

$$t_{O_{aj}}^- = \max \{t_{O_{aj+1}}^- + P_{O_{aj+1}}^-, s_{O_{aj,MS}}^- + P_{O_{aj,MS}}^-\} = \max \{t_{O_{aj+1}} + P_{O_{aj+1}}^-, t_{O_{aj,MS}} + P_{O_{aj,MS}}^-\}. \quad (3)$$

因此

$$\text{makespan} = \max \{j \in J | P_{O_{aj}}^- + \max \{s_{O_{aj-1}} + P_{O_{aj-1}}^-, s_{O_{aj,MP}} + P_{O_{aj,MP}}^-\} + \max \{t_{O_{aj+1}} + P_{O_{aj+1}}^-, t_{O_{aj,MS}} + P_{O_{aj,MS}}^-\}\}. \quad (4)$$

当阶段数为0时 $s = 0$, 当阶段数为 s 时 $s = 0$. 该方法得到的makespan即为从工件开始加工到加工结束的最长路径, 即求得的makespan与全解码时的实际值相等, 且计算时间复杂度由 $O_{(n*s)}$ 减小为 $O_{(n*2)}$. 若该方法仅针对调整工序的机器, 其计算时间将小于 $O_{(n*2)}$.

5 基于果蝇算法的混合优化方法

5.1 编码与解码

本文采用的编码方式为基于工件的序列编码, 如3, 5, 2, 4, 1, 即工件3最先安排机器最先加工, 其次是工件5, 最后是工件1. 解码采用的是基于排列的方

式^[25], 其中工件排序的第1阶段为给定的编码序列, 后续的阶段则按照先到先安排的原则进行, 对上一阶段的工件完成时间进行非降序排序以确定工件序列. 机器选择则是按照先到先加工的方式进行, 若存在几个工件的上阶段完工时间相同, 则随机挑选一个工件优先加工. 此外, 本文还采用了正向和逆向解码策略.

5.2 初始化方法

本文采用了GRASP(greedy randomized adaptive search procedure)-NEH_(x)^[29]的初始化方法, 并以 $x = 4$ 进行种群初始化. 为了保证种群的多样性, 在初始化的过程中, 需要对个体进行检验, 以防止种群中出现相同的个体.

5.3 嗅觉搜索与视觉搜索

嗅觉搜索和视觉搜索是FOA的两个核心步骤. 本文采用了包含交叉邻域结构、插入和交换邻域结构以及关键路径邻域结构的VNS, 对嗅觉搜索和视觉搜索进行改进. 其交叉方式主要通过“中心果蝇”对其他果蝇进行引导, 交叉方法如图4所示.

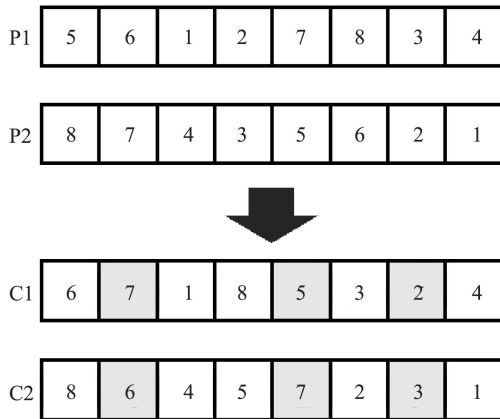


图4 交叉操作

Fig. 4 Cross operation

插入和交换邻域结构针对种群中的所有个体. 其中: 插入操作在编码序列中随机删除一个工件, 并在剩余的序列中选择一个位置, 插入该工件; 交换操作是在编码序列中随机挑选两个位置, 并交换该位置对应的工件. 若个体被更新, 将重复该方法; 若不更新, 则停止.

基于关键路径邻域结构的局部搜索的伪代码如算法1所示. 在搜索完成后, 通过对种群个体适应度值的非升序排序确定种群的“中心果蝇”. 最后, 利用锦标赛法选择适应度值较好的个体, 替换锦标赛中较差的个体, 并对该个体采用GRASP-NEH_(x)方法进行更新.

5.4 算法流程图

具体的求解算法流程图如图6所示.

6 实验结果与分析

本文所提混合优化方法(improved FOA, IFOA)采

用C++语言编程实现, 程序运行环境为: intel Core i5处理器, 主频1.6 GHz, 内存为8.00 GB. 为了使计算结果更具有有效性和一般性, 每个算例正向解码、逆向解码各独立运行5次. 其中种群规模为算例的工件个数, RunningTime为算法运行时间, Result为运行结果, MT为平均时间, LB为Benchmark, 即

$$MT = \frac{\sum_{i=1}^n \text{RunningTime}_i}{n}, \quad (5)$$

$$MT = \frac{\sum_{i=1}^n \text{Result}_i}{n}, \quad (6)$$

$$MT = \frac{C_{\max} - LB}{LB} \times 100\%. \quad (7)$$

算法1 基于关键路径的局部搜索伪代码

```
for (int i = 0; i < n; ++i) {
    产生一个临时个体P, 其中P = Pi;
    do {
        Pi = P;
        P -> getCriticalpath(); 获取关键路径
        switch(rand(0, 3)) {
            case(1): 同阶段机器间的工序插入; break;
            case(2): 同阶段机器间的工序交换; break;
        };
    } while(Pi != P);
} end for
```

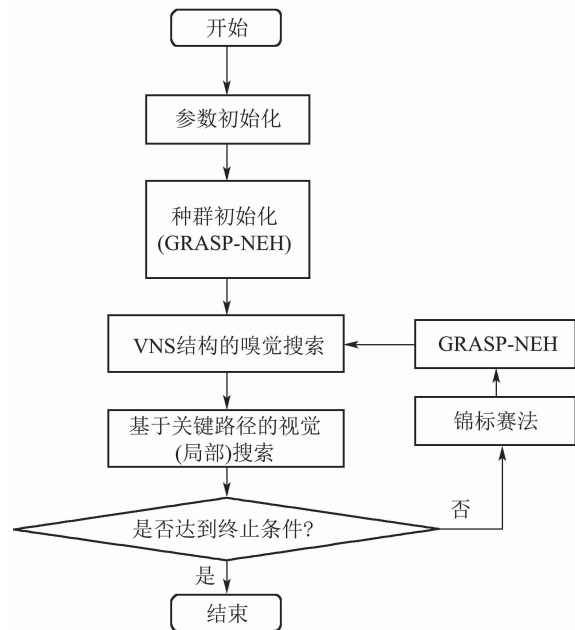


图6 算法流程

Fig. 6 Algorithm flow

6.1 新邻域结构的有效性验证

参照文献[30], 选择了Carrier测试集^[31]中较为复

杂的j10c10c1至j10c10c6案例进行有效性验证, 对每个测试案例的运行时间不超过60 s, 测试10次, 并统计最优解 $C_{\max}$ 、平均值AVG和偏差值PD. 其中: 未加入关键路径邻域结构的果蝇算法记为FOA, 加入关键路径邻域结构的果蝇算法记为IFOA. 实验结果如表2所示.

表2 新邻域结构的有效性验证

Table 2 Validation of the new neighborhood structure

测试集	LB	FOA			IFOA		
		$C_{\max}$	AVG	PD	$C_{\max}$	AVG	PD
j10c10c1	113	115	115	1.77%	114	114	0.88%
j10c10c2	116	116	117.5	0.00%	116	116.5	0.00%
j10c10c3	98	116	116.5	18.37%	116	116.5	18.37%
j10c10c4	103	120	120.5	16.50%	119	119.5	15.53%
j10c10c5	121	126	126.5	4.13%	125	126	3.31%
j10c10c6	97	105	105.5	8.25%	105	105.5	8.25%

6.2 快速评估方法的高效性验证

为了验证本文所提快速评估方法的高效性, 本文设计了阶段数为{5, 10, 15, 20}, 工件数为{10, 20, 30}的小规模算例12个和工件数为{40, 80, 160}的大规模算例12个. 其中: 每个阶段的可选机器数为[2, 4], 加工时间为[10, 80]. 对每个算例计算10次, 算法每次执行100000次, 并采用快速评估中耗时最长的情况(对第2部分的工件全部计算)与全解码的运行时间进行对比. 记录快速评估和全解码的10次运行时间的平均值, 如图7~8所示, 横轴表示加工的阶段数, 纵轴表示一次解码的时间.

对比图7~8可知, 快速评估的运行速度较全解码快. 当工件数量一定时, 快速评估的速度不受加工阶段数的影响, 而全解码随着阶段数的增加, 运行时间也将增加.

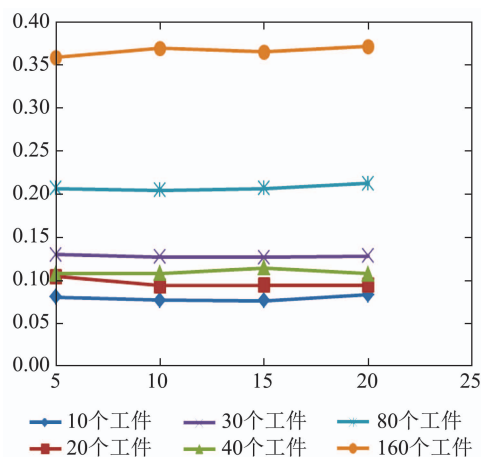


图7 快速评估的运行结果

Fig. 7 Result of rapid evaluation method

由表2可知, 在6个算例中, IFOA算法实验取得的最优解和平均值较优, 证明了提出的邻域结构可以有效平衡算法的局部搜索和全局搜索性能, 并提高了算法搜索更优解的概率. 所提算法运行10次的结果中 $C_{\max}$ 与PD值相差不大, 体现了IFOA算法在求解该类问题时的稳定性较好.

时间与快速评估时间的比值. 当阶段数增加时, 快速评估的计算时间远远低于全解码. 因此, 该方法在保证求解结果准确性的同时, 还能有效地提高了计算速度, 且阶段数越大该方法效果越明显.

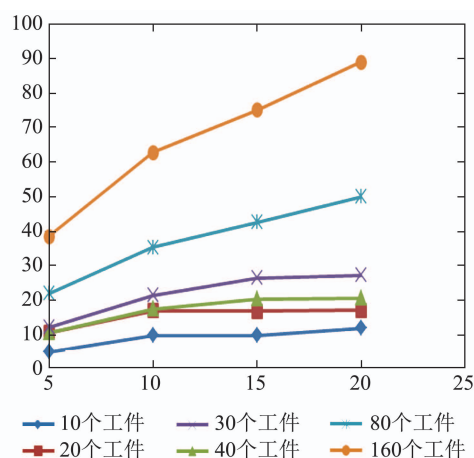


图8 全解码的运行结果

Fig. 8 Result of normal method

6.3 混合优化方法的有效性验证

6.3.1 Liao测试集大规模算例实验结果

本文采用了Liao等^[32]的10个较大规模测试集进行测试, 每个算例运行10次, 记录实验结果并计算值. 将IFOA算法与目前较好的算法: 改进差分进化算法^[8](improved differential evolutionary algorithm, IDE)、混合多目标人工蜂群算法^[33](ABC)以及改进生物地理学优化算法^[34](improved biogeography-based optimization algorithm, IBBO)进行实验结果对比. 并在进行IFOA迭代次数的设定时, 参照文献[8]的IDE迭代次数设定, 将IFOA的迭代次数设为200. 实验结果如表3所示.

如图9所示, 横轴表示加工阶段数, 纵轴为全解码

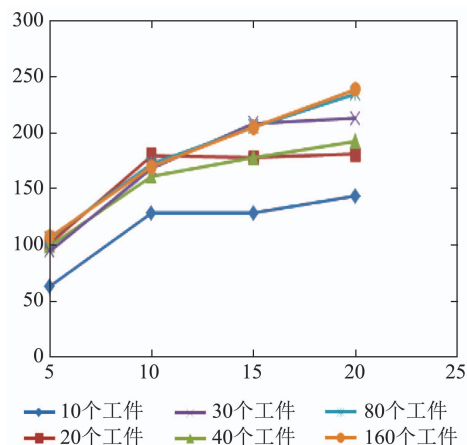


图9 全解码与快速评估的比值结果

Fig. 9 Ratio result of rapid normal to evaluation method

由表3可知,新的邻域结构提高了IFOA算法的搜索能力,其求得的最优解和解的平均值均优于对比算法.实验结果表明,在求解较大规模问题时,IFOA算法的搜索能力更强,最优解的质量更优,从而验证了IFOA算法以及所提邻域结构的有效性和优越性.

6.3.2 Carlier测试集a类和b类算例的实验结果

鉴于a类与b类问题较为简单,每个算例运行时间不超过60 s,记录算法运行时间(保留整数),并计算算例的平均运行时间MT值.其中:ForwardTime表示正向解码时间,BackwardTime表示逆向解码时间,实验结果如表4所示.

从表4实验结果可知,47个案例均能找到下界,且不难发现本文提出的算法在初始化阶段就能找到大多数算例的最优解,从而说明本文提出的IFOA算法的有效性.

6.3.3 Carlier测试集c类和d类算例的实验结果

由于Carlier测试集c类和d类问题的复杂度较高,在进行实验时每个算例运行时间不超过60 s,并记录

最优值.将本文IFOA算法与改进离散人工蜂群算法^[29](improved discrete artificial bee colony algorithm, IDABC)、改进粒子群算法^[35](improved particle swarm optimization algorithm, IPOS)以及改进灰狼算法^[30](first arrive first process-improved grey wolf optimization, FF-IGWO)进行实验结果对比.其中前两个算法采用原有文献结果,而FF-IGWO文献中使用了3种解码方式进行算例测试,因此采用与本文相同解码方式的结果,与本文方法进行对比.实验结果如表5所示.由表5可知,针对上面这30个实例,在最好解方面(即4个算法中的最好解),所提IFOA得到了28个最好解,IPOS求得了27个最好解,IDABC求得了22个最好解,FF-IGWO求得了25个最好解,其中IFOA搜索到了j10c10c2的最优解116,以及j10c10c4的更好解119.图10展示了测试算例j10c10c4目标值为119的甘特图.从实验结果可以看出,本文提出的方法可以获得较好的结果.

7 总结与展望

针对混合流水车间的问题特性,构建了同阶段机器间工序交换的新邻域结构,引入了FJSP中的N7和K-insertion邻域结构.并针对关键路径的邻域结构在搜索过程中耗时较大的缺陷,设计了一种快速评估方法,提高了求解效率.最终,将关键路径邻域结构与VNS结合,提出了一种基于FOA的混合优化方法,并通过实验证明了所提方法的有效性和优越性.

目前的研究多半是基于智能优化算法的研究,缺乏对车间特点的研究.因此,本文提出了以下展望:

- 1) 基于HFSP的特性,对关键路径上机器间的邻域结构进行研究,从而减少不必要的搜索,提高搜索效率;
- 2) 针对不同特性的HFSP进行解空间的裁剪,以提高算法的求解效率.

表3 Liao10个大规模算例实验结果

Table 3 The experimental results of 10 large-scale benchmark of Liao

测试集	IDE		LABC		IBBO		IFOA	
	$C_{\max}$	AVG	$C_{\max}$	AVG	$C_{\max}$	AVG	$C_{\max}$	AVG
j30c5e1	474	477.8	467	468.83	474	479.2	464	465.7
j30c5e2	616	618.1	616	616	616	617.6	616	616
j30c5e3	602	605.2	596	599.2	610	614.1	595	597.4
j30c5e4	575	578.7	571	572.4	577	582	566	568.7
j30c5e5	605	607.2	603	604.4	609	616.9	602	603.7
j30c5e6	612	615.3	607	607.8	615	620	603	603.6
j30c5e7	630	634.1	626	627.4	629	635.5	626	626
j30c5e8	685	687.5	678	678	685	693.1	675	676.4
j30c5e9	651	655	646	646	654	662.5	644	645.3
j30c5e10	594	597.3	580	582	596	605.4	578	580.5

表4 Carlier测试集a类和b类算例实验结果

Table 4 The experimental results of Carlier's a and b benchmark

测试集	LB	$C_{\max}$	ForwardTime/s				MT		BackwardTime/s				MT	
j10c5a2	88	88	0.002	0.009	0.001	0.008	0.002	0.004	0.028	0.024	0.002	0.011	0.01	0.015
j10c5a3	117	117	0.002	0.018	0.011	0.017	0.002	0.011	0.007	0.023	0.013	0.007	0.005	0.011
j10c5a4	121	121	0.003	0.026	0.004	0.006	0.005	0.025	0.013	0.027	0.004	0.01	0.154	0.042
j10c5a5	122	122	0.043	0.027	0.023	0.004	0.016	0.017	0.005	0.027	0.009	0.006	0.007	0.011
j10c5a6	110	110	0.041	0.056	0.029	0.022	0.122	0.033	0.002	0.006	0.023	0.008	0.023	0.012
j10c5b1	130	130	0.004	0.002	0.013	0.002	0.012	0.005	0.002	0.002	0.001	0.002	0.005	0.002
j10c5b2	107	107	0.002	0.002	0.002	0.007	0.002	0.003	0.002	0.001	0.002	0.002	0.004	0.002
j10c5b3	109	109	0.008	0.002	0.008	0.005	0.011	0.007	0.012	0.004	0.013	0.004	0.003	0.007
j10c5b4	122	122	0.015	0.005	0.02	0.028	0.014	0.011	0.004	0.004	0.004	0.004	0.013	0.006
j10c5b5	153	153	0.004	0.002	0.002	0.003	0.001	0.002	0.002	0.002	0.001	0.002	0.001	0.002
j10c5b6	115	115	0.002	0.012	0.002	0.001	0.001	0.003	0.001	0.002	0.002	0.002	0.003	0.002
j10c10a1	139	139	0.052	0.014	0.006	0.016	0.047	0.039	0.067	0.007	0.073	0.039	0.068	0.051
j10c10a2	158	158	0.163	0.043	0.321	0.442	0.157	0.144	0.054	0.098	0.076	0.045	0.042	0.063
j10c10a3	148	148	0.011	0.064	0.026	0.019	0.01	0.109	0.265	0.141	0.237	0.269	0.049	0.192
j10c10a4	149	149	0.014	0.046	0.053	0.071	0.056	0.031	0.005	0.003	0.008	0.051	0.003	0.014
j10c10a5	148	148	0.043	0.066	0.025	0.043	0.055	0.028	0.018	0.006	0.014	0.003	0.007	0.01
j10c10a6	146	146	0.077	0.078	0.005	0.307	0.093	0.076	0.084	0.015	0.003	0.054	0.043	0.04
j10c10b1	163	163	0.003	0.019	0.003	0.005	0.007	0.006	0.003	0.004	0.003	0.004	0.006	0.004
j10c10b2	157	157	0.059	0.059	0.062	0.061	0.01	0.038	0.048	0.043	0.023	0.01	0.007	0.026
j10c10b3	169	169	0.014	0.003	0.005	0.003	0.017	0.008	0.008	0.002	0.003	0.006	0.014	0.007
j10c10b4	159	159	0.003	0.002	0.008	0.011	0.026	0.009	0.003	0.01	0.003	0.006	0.015	0.007
j10c10b5	165	165	0.016	0.022	0.044	0.017	0.005	0.02	0.052	0.008	0.009	0.005	0.02	0.019
j10c10b6	165	165	0.012	0.002	0.005	0.015	0.022	0.009	0.017	0.003	0.003	0.007	0.003	0.007
j15c5a1	178	178	0.076	0.069	0.007	0.072	0.091	0.05	0.014	0.02	0.087	0.04	0.023	0.037
j15c5a2	165	165	0.016	0.027	0.017	0.006	0.01	0.023	0.004	0.036	0.05	0.022	0.038	0.03
j15c5a3	130	130	0.015	0.028	0.023	0.029	0.012	0.042	0.038	0.108	0.089	0.046	0.036	0.063
j15c5a4	156	156	0.004	0.076	0.068	0.031	0.019	0.084	0.184	0.102	0.249	0.005	0.103	0.129
j15c5a5	164	164	0.023	0.027	0.004	0.029	0.081	0.073	0.034	0.103	0.336	0.011	0.083	0.113
j15c5a6	178	178	0.01	0.018	0.004	0.005	0.028	0.063	0.076	0.08	0.073	0.267	0.073	0.114
j15c5b1	170	170	0.003	0.004	0.006	0.014	0.012	0.008	0.011	0.008	0.006	0.008	0.004	0.007
j15c5b2	152	152	0.007	0.004	0.019	0.008	0.023	0.01	0.005	0.005	0.009	0.006	0.011	0.007
j15c5b3	157	157	0.026	0.017	0.008	0.004	0.011	0.016	0.038	0.024	0.016	0.005	0.008	0.018
j15c5b4	147	147	0.078	0.016	0.078	0.004	0.022	0.041	0.026	0.019	0.004	0.08	0.078	0.041
j15c5b5	166	166	0.076	0.021	0.033	0.074	0.078	0.043	0.047	0.004	0.079	0.008	0.006	0.029
j15c5b6	175	175	0.006	0.021	0.014	0.011	0.03	0.013	0.019	0.008	0.004	0.005	0.008	0.009
j15c10a1	236	236	0.006	0.031	0.023	0.046	0.027	0.02	0.026	0.006	0.016	0.007	0.008	0.013
j15c10a2	200	200	0.147	0.196	0.185	0.189	0.187	0.02	0.149	0.143	0.041	0.085	0.15	0.114
j15c10a3	198	198	0.136	0.132	0.151	0.072	0.183	0.147	0.019	0.007	0.068	0.006	0.012	0.022
j15c10a4	225	225	0.223	0.02	0.016	0.177	0.034	0.079	0.072	0.057	0.026	0.128	0.01	0.059
j15c10a5	182	182	0.16	0.015	0.051	0.011	0.153	0.076	0.143	0.018	0.146	0.089	0.019	0.083
j15c10a6	200	200	0.013	0.046	0.085	0.04	0.082	0.081	0.147	0.151	0.031	0.035	0.012	0.075
j15c10b1	222	222	0.017	0.018	0.008	0.012	0.029	0.064	0.14	0.04	0.134	0.031	0.058	0.081
j15c10b2	187	187	0.028	0.006	0.055	0.038	0.024	0.039	0.082	0.062	0.006	0.051	0.035	0.047
j15c10b3	222	222	0.011	0.025	0.014	0.006	0.016	0.025	0.066	0.024	0.011	0.042	0.038	0.036
j15c10b4	221	221	0.026	0.006	0.007	0.015	0.031	0.02	0.006	0.006	0.061	0.01	0.032	0.023
j15c10b5	200	200	0.152	0.13	0.146	0.043	0.272	0.153	0.14	0.081	0.146	0.23	0.187	0.157
j15c10b6	219	219	0.056	0.136	0.131	0.022	0.02	0.052	0.059	0.018	0.062	0.006	0.013	0.032

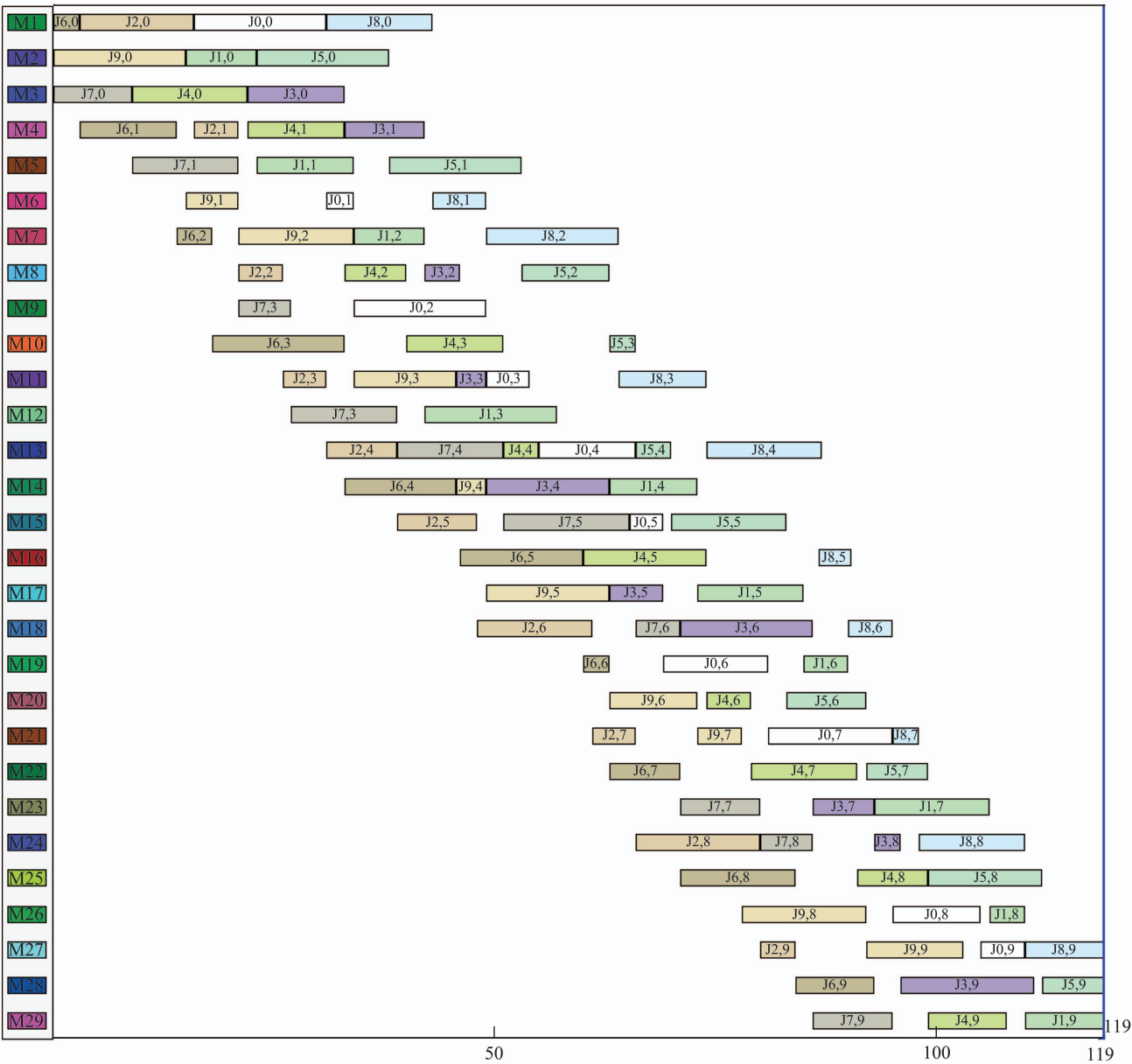


图 10 Carlier测试集j10c10c4实例的甘特图
Fig. 10 Gantt chart of the Carlier's j10c10c4 benchmark

表 5 Carlier测试集c类和d类算例实验结果

Table 5 The experimental results of Carlier's c and d benchmark

测试集	LB	IDABC		IPOS		FF-IGWO		IFOA	
		$C_{\max}$	PD	$C_{\max}$	PD	$C_{\max}$	PD	$C_{\max}$	PD
j10c5c1	68	68	0.00%	68	0.00%	68	0.00%	68	0.00%
j10c5c2	74	74	0.00%	74	0.00%	74	0.00%	74	0.00%
j10c5c3	71	72	1.41%	71	0.00%	72	1.41%	71	0.00%
j10c5c4	66	66	0.00%	66	0.00%	66	0.00%	66	0.00%
j10c5c5	78	78	0.00%	78	0.00%	78	0.00%	78	0.00%
j10c5c6	69	69	0.00%	69	0.00%	69	0.00%	69	0.00%
j10c5d1	66	66	0.00%	66	0.00%	66	0.00%	66	0.00%
j10c5d2	73	73	0.00%	73	0.00%	73	0.00%	73	0.00%

(转下一页)

(接上一页)

测试集	LB	IDABC		IPOS		FF-IGWO		IFOA	
		$C_{\max}$	PD	$C_{\max}$	PD	$C_{\max}$	PD	$C_{\max}$	PD
j10c5d3	64	64	0.00%	64	0.00%	64	0.00%	64	0.00%
j10c5d4	70	70	0.00%	70	0.00%	70	0.00%	70	0.00%
j10c5d5	66	66	0.00%	66	0.00%	66	0.00%	66	0.00%
j10c5d6	62	62	0.00%	62	0.00%	62	0.00%	62	0.00%
j10c10c1	113	115	1.77%	113	0.00%	114	0.88%	114	0.88%
j10c10c2	116	119	2.59%	117	0.86%	117	0.86%	116	0.00%
j10c10c3	98	116	18.37%	116	18.37%	116	18.37%	116	18.37%
j10c10c4	103	120	16.50%	120	16.50%	120	16.50%	119	15.53%
j10c10c5	121	125	3.31%	125	3.31%	125	3.31%	125	3.31%
j10c10c6	97	106	9.28%	106	9.28%	105	8.25%	105	8.25%
j15c5c1	85	85	0.00%	85	0.00%	85	0.00%	85	0.00%
j15c5c2	90	90	0.00%	90	0.00%	90	0.00%	90	0.00%
j15c5c3	87	87	0.00%	87	0.00%	87	0.00%	87	0.00%
j15c5c4	89	89	0.00%	89	0.00%	89	0.00%	89	0.00%
j15c5c5	73	74	1.37%	73	0.00%	74	1.37%	74	1.37%
j15c5c6	91	91	0.00%	91	0.00%	91	0.00%	91	0.00%
j15c5d1	167	167	0.00%	167	0.00%	167	0.00%	167	0.00%
j15c5d2	82	84	2.44%	84	2.44%	84	2.44%	84	2.44%
j15c5d3	77	82	6.49%	82	6.49%	82	6.49%	82	6.49%
j15c5d4	61	84	37.70%	84	37.70%	84	37.70%	84	37.70%
j15c5d5	67	79	17.91%	79	17.91%	79	17.91%	79	17.91%
j15c5d6	79	81	2.53%	81	2.53%	81	2.53%	81	2.53%

参考文献:

- [1] LI Yingli, LI Xinyu, GAO Liang. Review on hybrid flow shop scheduling problems. *Chinese Journal of Mechanical Engineering*, 2020, 31(23): 2798 – 2813, 2828.
(李颖俐, 李新宇, 高亮. 混合流水车间调度问题研究综述. 中国机械工程, 2020, 31(23): 2798 – 2813, 2828.)
- [2] WANG Ling, ZHOU Gang, XU Ye, et al. Advances in the study on hybrid flow-shop scheduling. *Control and Instruments in Chemical Industry*, 2011, 38(1): 1 – 8, 22.
(王凌, 周刚, 许烨, 等. 混合流水线调度研究进展. 化工自动化及仪表, 2011, 38(1): 1 – 8, 22.)
- [3] LEI Deming, WANG Tian. A novel shuffled frog-leaping algorithm for low carbon hybrid flow shop scheduling. *Control and Decision*, 2021, 36(1): 241 – 248.
(雷德明, 王甜. 基于改进蛙跳算法的分布式两阶段混合流水车间调度. 控制与决策, 2021, 36(1): 241 – 248.)
- [4] WANG Shengyao, WANG Ling, FANG Chen, et al. Advances in estimation of distribution algorithms. *Control and Decision*, 2012, 27(7): 961 – 966, 974.
(王圣尧, 王凌, 方晨, 等. 分布估计算法研究进展. 控制与决策, 2012, 27(7): 961 – 966, 974.)
- [5] ZHENG Xiaocao, GONG Wenyin. An improved artificial bee colony algorithm for fuzzy flexible job-shop scheduling problem. *Control Theory & Applications*, 2020, 37(6): 1284 – 1292.
(郑小操, 龚文引. 改进人工蜂群算法求解模糊柔性车间调度问题. 控制理论与应用, 2020, 37(6): 1284 – 1292.)
- [6] LIAO Zuowen, GONG Wenyin, WANG Ling. A hybrid swarm intelligence with improved ring topology for nonlinear equations. *Scientia Sinica Informationis*, 2020, 50(3): 396 – 407.
(廖作文, 龚文引, 王凌. 基于改进环拓扑混合群体智能算法的非线性方程组多根联解. 中国科学: 信息科学, 2020, 50(3): 396 – 407.)
- [7] WANG S Y, WANG L, LIU M, et al. An enhanced estimation of distribution algorithm for solving hybrid flow-shop scheduling problem with identical parallel machines. *The International Journal of Advanced Manufacturing Technology*, 2013, 68(2): 9 – 12.
- [8] ZHANG Yuan, TAO Yifei, WANG Jiamian. An improved differential evolutionary algorithm to solve the hybrid flow-shop scheduling problems. *Chinese Journal of Mechanical Engineering*, 2021, 32(6): 714 – 720.
(张源, 陶翼飞, 王加冕. 改进差分进化算法求解混合流水车间调度问题. 中国机械工程, 2021, 32(6): 714 – 720.)
- [9] PAN Q K, WANG L, LI J Q, et al. A novel discrete artificial bee colony algorithm for the hybrid flowshop scheduling problem with makespan minimisation. *Omega*, 2014, 45(6): 42 – 56.
- [10] LI Y L, LI X Y, GAO L, et al. A discrete artificial bee colony algorithm for distributed hybrid flowshop scheduling problem with sequence dependent setup times. *International Journal of Production Research*, 2020, 59(13): 3880 – 3899.
- [11] LI Y L, LI X Y, GAO L, et al. An improved artificial bee colony algorithm for distributed heterogeneous hybrid flowshop scheduling problem with sequence-dependent setup times. *Computers and Industrial Engineering*, 2020, DOI: 10.1016/j.cie.2020.106638.
- [12] LU C, GAO L, PAN Q K, et al. A multi-objective cellular grey wolf optimizer for hybrid flowshop scheduling problem considering noise pollution. *Applied Soft Computing Journal*, 2018, 75(2): 728 – 749.
- [13] ZHANG C J, TAN J W, PENG K K, et al. A discrete whale swarm algorithm for hybrid flow-shop scheduling problem with limited

- buffers. *Robotics and Computer-Integrated Manufacturing*, 2021, DOI: 10.1016/j.rcim.2020.102081.
- [14] DONG Jun, YE Chunming. Collaborative optimization of interval number reentrant hybrid flow shop scheduling and preventive maintenance. *Control and Decision*, 2021, 36(11): 1 – 10.
(董君, 叶春明. 区间数可重入混合流水车间调度与预维护协同优化. 控制与决策, 2021, 36(11): 1 – 10.)
- [15] LEI Deming, YANG Dongjing. A novel shuffled frog-leaping algorithm for low carbon hybrid flow shop scheduling. *Control and Decision*, 2020, 35(6): 1329 – 1337.
(雷德明, 杨冬婧. 基于新型蛙跳算法的低碳混合流水车间调度. 控制与决策, 2020, 35(6): 1329 – 1337.)
- [16] LEI Deming, WANG Tian. An improved shuffled frog leaping algorithm for the distributed two-stage hybrid flow shop scheduling. *Control and Decision*, 2021, 36(1): 241 – 248.
(雷德明, 王甜. 基于改进蛙跳算法的分布式两阶段混合流水车间调度. 控制与决策, 2021, 36(1): 241 – 248.)
- [17] PAN Quanke, GAO Liang, LI Xinyu. *Flowshop Scheduling and Its Optimization Algorithm*. Wuhan: Huazhong University of Science and Technology Press, 2011.
(潘全科, 高亮, 李新宇. 流水车间调度及其优化算法. 武汉: 华中科技大学出版社, 2011.)
- [18] WANG Ling, ZHENG Xiaolong. Advances in fruit fly optimization algorithms. *Control Theory & Applications*, 2017, 34(5): 557 – 563.
(王凌, 郑晓龙. 果蝇优化算法研究进展. 控制理论与应用, 2017, 34(5): 557 – 563.)
- [19] WANG Ling, SHENG Jingnan, WANG Shengyao, et al. Advances in co-evolutionary algorithms. *Control and Decision*, 2015, 30(2): 193 – 202.
(王凌, 沈婧楠, 王圣尧, 等. 协同进化算法研究进展. 控制与决策, 2015, 30(2): 193 – 202.)
- [20] WANG L, SHI Y, LIU S. An improved fruit fly optimization algorithm and its application to joint replenishment problems. *Expert Systems with Applications*, 2015, 42(9): 4310 – 4323.
- [21] ZHENG X L, WANG L. A two-stage adaptive fruit fly optimization algorithm for unrelated parallel machine scheduling problem with additional resource constraints. *Expert Systems with Applications*, 2016, 65(12): 28 – 39.
- [22] SHAO Z S, PI D C, SHAO W S. Hybrid enhanced discrete fruit fly optimization algorithm for scheduling blocking flow-shop in distributed environment. *Expert Systems with Applications*, 2020, 145(5): 1 – 17.
- [23] WANG Ling, ZHENG Jie, WANG Jingjing. A hybrid discrete fruit fly optimization algorithm for distributed permutation flowshop scheduling with interval data. *Control and Decision*, 2020, 35(4): 930 – 936.
(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. 控制与决策, 2020, 35(4): 930 – 936.)
- [24] WANG L, ZHENG X L. A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem. *Swarm and Evolutionary Computation*, 2018, 38(2): 54 – 63.
- [25] DUAN J H, CHEN Q D, SUN W Q, et al. A multi-swarm fruit fly optimization algorithm to minimize makespan for the hybrid flow-shop problem. *Proceedings of the 36th Chinese Control Conference*. Dalian, China: IEEE, 2017: 2796 – 2800.
- [26] PENG K K, PAN Q K, GAO L, et al. A multi-start variable neighbourhood descent algorithm for hybrid flowshop rescheduling. *Swarm and Evolutionary Computation*, 2019, 45(3): 92 – 112.
- [27] ZHANG C Y, LI P G, GUAN Z L, et al. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. *Computers and Operations Research*, 2005, 34(11): 3229 – 3242.
- [28] MACTROLILLI M, GAMBARDELLA L M. Effective neighbourhood functions for the flexible job shop problem. *Journal of Scheduling*, 2000, 3(1): 3 – 20.
- [29] CUI Z, GU X S. An improved discrete artificial bee colony algorithm to minimize the makespan on hybrid flow shop problems. *Neuro-computing*, 2015, 148(1): 248 – 259.
- [30] SHI Weiguo, SONG Cunli. Improved grey wolf optimization to solve the hybrid flow shop scheduling problem. *Computer Integrated Manufacturing Systems*, 2021, 27(11): 3196 – 3208.
(时维国, 宋存利. 求解混合流水车间调度问题的改进灰狼算法. 计算机集成制造系统, 2021, 27(11): 3196 – 3208.)
- [31] JACQUES C, EMMANUEL N. An exact method for solving the multi-processor flow-shop. *RAIRO-Operations Research*, 2000, 34(1): 1 – 25.
- [32] LIAO C, EVI T, CHUNG T. An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem. *Applied Soft Computing*, 2012, 12(6): 1755 – 1764.
- [33] LI J Q, HAN Y Q. A hybrid multi-objective artificial bee colony algorithm for flexible task scheduling problems in cloud computing system. *Cluster Computing*, 2020, 23(12): 2483 – 2499.
- [34] LI Z P, GU X S. Improved biogeography-based optimization algorithm used in solving hybrid flow shop scheduling problem. *CIESC Journal*, 2016, 67(3): 751 – 757.
- [35] GU Wenbin, LI Yuxin, QIAN Yuhui, et al. Improved particle swarm optimization algorithm with hormone modulation mechanism for solving hybrid flow-shop scheduling problem with identical parallel machine. *Computer Integrated Manufacturing Systems*, 2021, 27(14): 2858 – 2871.
(顾文斌, 李育鑫, 钱煜晖, 等. 基于激素调节机制IPSO算法的相同并行机混合流水车间调度问题. 计算机集成制造系统, 2021, 27(14): 2858 – 2871.)

作者简介:

周永强 硕士研究生, 研究方向为智能优化算法与车间调度, E-mail: 2532868784@qq.com;

王翠雨 博士研究生, 研究方向为智能优化方法及其应用, E-mail: 76325434@qq.com;

李颖俐 博士研究生, 研究方向为智能优化算法与车间调度, E-mail: lyl.elena@163.com;

李新宇 教授, 博士生导师, 博士, 研究方向为智能制造系统、车间调度、智能优化与机器学习, E-mail: lixinyu@mail.hust.edu.cn.