



计算机科学

COMPUTER SCIENCE

面向异步混合流水车间排产的混合禁忌搜索遗传优化算法

王思彤, 林荣恒

引用本文

王思彤, 林荣恒. 面向异步混合流水车间排产的混合禁忌搜索遗传优化算法[J]. 计算机科学, 2025, 52(4): 271-279.

WANG Sitong, LIN Rongheng. Improved Genetic Algorithm with Tabu Search for Asynchronous Hybrid Flow Shop Scheduling [J]. Computer Science, 2025, 52(4): 271-279.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[图着色问题禁忌搜索改进算法](#)

Improved Algorithm for Tabu Search of Graph Coloring Problems

计算机科学, 2022, 49(11A): 211000128-5. <https://doi.org/10.11896/jsjcx.211000128>

[基于改进准深度算法的诊断策略优化方法](#)

Diagnosis Strategy Optimization Method Based on Improved Quasi Depth Algorithm

计算机科学, 2022, 49(6A): 729-732. <https://doi.org/10.11896/jsjcx.210700076>

[面向通用航空器运行排班及维修的策略优化](#)

Optimization of Scheduling and Maintenance Strategy for Navigation Aircraft Operation

计算机科学, 2020, 47(11A): 632-637. <https://doi.org/10.11896/jsjcx.200600053>

[基于改进变邻域搜索的数控铣床路径优化](#)

Path Optimization in CNC Cutting Machine Based on Modified Variable Neighborhood Search

计算机科学, 2020, 47(10): 233-239. <https://doi.org/10.11896/jsjcx.190800035>

[优化能耗的可变电压禁忌任务调度算法](#)

Variable Voltage Tabu Task Scheduling Algorithm for Optimizing Energy Consumption

计算机科学, 2010, 37(10): 287-290.

面向异步混合流水车间排产的混合禁忌搜索遗传优化算法

王思彤 林荣恒

北京邮电大学计算机学院(国家示范性软件学院) 北京 100876

(wangsitong0814@bupt.edu.cn)

摘要 相比于传统流水线,混合流水车间具有更高的灵活性,能适应多变的生产场景,但其排产方案的求解复杂度更高,是现代实际制造系统中的常见问题。针对群智能进化算法在解决该问题时计算难度大且搜索效率不高的问题,以最小化总完工时间为优化目标,提出了一种混合禁忌搜索遗传优化算法。该算法根据排产问题中所有工件具有相同生产工艺、工件数量多、各阶段并行机不同速的特点,采用了基于首阶段工件顺序的单层编码、考虑机器选择三层优先级的解码方法、多种遗传算子和禁忌搜索算子,具有更加优秀的搜索性能,在保证解质量的基础上提高了算法的收敛速度。最后,通过40个算例和实际应用案例评估算法性能,并将其与其他算法进行比较。实验结果表明,所提出的算法在求解中规模算例、大规模算例和加工车间案例时表现优秀,排产结果的完工时间平均缩短了10.71%,算法达到最优解所需的迭代次数减少了25.72%,运行时间缩短了10.79%。

关键词:排产优化;改进遗传算法;禁忌搜索;混合流水车间调度;异步并行机

中图分类号 TP301

Improved Genetic Algorithm with Tabu Search for Asynchronous Hybrid Flow Shop Scheduling

WANG Sitong and LIN Rongheng

College of Computer Science(National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract Compared with the traditional assembly line, the hybrid flow shop has higher flexibility and can adapt to the changing production scenarios. However, the solution complexity of its scheduling scheme is higher, which is a common problem in modern actual manufacturing systems. Aiming at the problems of high computational difficulty and low search efficiency of swarm intelligence evolutionary algorithm in solving this problem, an improved genetic algorithm with tabu search is proposed to minimize the total completion time. According to the characteristics of all workpieces in the scheduling problem with the same production process, the large number of workpieces, and the different speeds of parallel machines in each stage, the proposed algorithm adopts a single-layer coding based on the first-stage workpiece order, a decoding method considering the three-layer priority of machine selection, a variety of genetic operators and tabu search operators. It has better search performance and improves the convergence speed of the algorithm on the basis of ensuring the quality of the solution. Finally, the performance of the algorithm is evaluated by 40 instances and workshop cases, and is compared with other algorithms. The experimental results show that the proposed algorithm performs well in solving medium-scale instances, large-scale instances and processing workshop cases. The completion time of the scheduling results is shortened by 10.71% on average. The number of iterations required for the algorithm to reach the optimal solutions is reduced by 25.72%, and the running time is shortened by 10.79%.

Keywords Scheduling optimization, Improved genetic algorithm, Tabu search, Hybrid flow shop scheduling, Asynchronous parallel machine

1 引言

随着新一代技术的飞速发展,制造业正朝着智能化、自动化等方向转变,单一的生产模式已无法满足多样化的需求。混合流水车间与传统流水生产线生产的主要区别在于其在

各个工序阶段允许存在多台能够并行工作的机器,进而可以提高车间的生产效率,增加产能。混合流水车间模型和方法不仅可以应用在制造领域,还可以应用于港口调度、医疗服务、物流配送等新兴领域^[1]。

面对不同的机器、工件和约束差异,合理安排产品的生产

到稿日期:2024-06-05 返修日期:2024-09-14

基金项目:国家重点研发计划(2021YFB3300700)

This work was supported by the National Key Research and Development Program of China(2021YFB3300700).

通信作者:林荣恒(rhlin@bupt.edu.cn)

顺序和选择合适的机器进行生产,是混合流水线车间排产问题的关键。国内外学者对优化排产问题进行了大量研究。Rauf等^[2]考虑总完工时间和空闲时间因素,使用改进的智能多目标算法对装配线的排产问题进行求解。Mosadegh等^[3]设计了一个改进混合 Q-Learning 的超模拟退火算法对多工位混流装配生产线排产进行优化。Zhou等^[4]结合递归调用提出了一种智能优化算法,用于解决具有优先级订单的车间排产问题。Lv等^[5]提出了一种改进的多目标粒子群算法来求解混合多处理机的车间调度问题。Li等^[6]考虑车间的邻域结构,使用改进人工蜂群算法解决带有准备时间的分布式混合流水车间排产问题。Lu等^[7]针对带有噪声干扰的混流车间排产问题,设计了一种改进的灰狼求解算法。Dong等^[8]提出改进的鲸鱼算法来对可重入的混流车间排产问题进行优化。Zhou等^[9]结合果蝇算法和变领域搜索的优化方法提高混合流水车间排产问题的搜索效率。Zhang等^[10]针对混流装配生产线的能耗平衡和联合排产,提出一种基于元胞策略的遗传算法。Defersha等^[11]提出了一种混合整数线性规划模型的遗传算法,用于解决带有输送机的装配线生产问题。

目前大多数研究基于同速混合流水车间场景,并根据特殊工艺约束、机器特征约束、不确定加工时间等对问题进行扩展。但实际生产环境中,混合流水车间同一加工阶段的并行机,可能因为机器批次、机器损耗等因素影响,而对生产产品的加工时间不一致^[12]。异步并行机的加工能力不同,排产过程中对机器选择更具有灵活性,求解复杂度更高,难度更大。Lopes等^[13]使用迭代分解算法对异步混流装配生产线排产问题进行多目标均衡优化。Adreu等^[14]为解决异步并行机的调度问题,设计了一种结合元启发式生成序列的模拟退火算法。Xuan等^[15]提出混合遗传的模拟退火算法,用于解决异步并行机的多阶段混合流水车间排产问题。

目前求解混合流水车间调度问题的主要方法仍是群智能优化算法。遗传算法作为被广泛应用的群智能优化算法之一,其优秀的全局搜索及优化能力能最大化减小不同生产场景对结果的影响,在有限的时间内提出较优的排产方案;但其在大规模场景下解空间大、迭代时间长、后期的搜索效率较低、解的改进效果不佳等问题。相比遗传算法,粒子群算法和模拟退火算法规则简单,实现容易且收敛速度快,但是容易陷入局部最优,对结果的改进效果不明显。变邻域搜索、禁忌搜索等局部搜索算法基于邻域结构对解进行优化,但是对邻域结构的设计具有局限性,全局搜索能力差。

综上所述,针对目前关于异步并行机混合流水车间排产的研究较少,且遗传算法的搜索效果不佳等问题,本文基于更接近实际场景中的异步混合流水车间排产问题设计模型,以优化最大完工时间为求解目标,根据模型特点设计染色体编码,使用多种选择、交叉、变异遗传算子提高种群多样性,并利用禁忌搜索优秀的局部寻优能力改进个体解,优化算法的搜索效率。最后,运用排产实例进行对比实验,证明提出的混合禁忌搜索遗传优化算法具有明显优势。

2 问题描述及定义

基于生产中异步并行机混合流水线车间的实际情况,

可以将排产问题描述为:生产流水线上包括 m 道工序阶段,每道工序存在并行的 $M_j (M_j \geq 1, j = 1, 2, \dots, m)$ 台机器。根据订单情况,有 n 个待加工的工件,其中每个工件加工均需要经过流水线上的 m 道工序,工件被加工至第 j 道工序时可以由该工序中异步并行的 M_j 台机器中的任意一台机器加工。混合流水车间排产过程见图 1。在已知工序数、机器数、待加工工件数和机器作业时间等因素的情况下,需要根据排产目标,合理安排工件的加工顺序并选择合适的并行机器,对 n 个工件排产方案的最大完工时间进行优化求解。

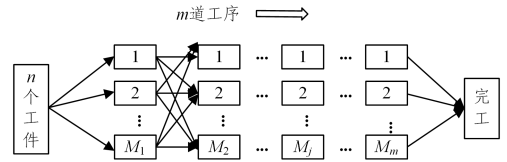


图 1 混合流水车间图示

Fig. 1 Illustration of hybrid flow shop

2.1 约束条件

为求解混合流水车间排产问题,设定如下约束条件:

- (1) 工件具有独立性,每个工件作为一个完整单元;
- (2) 禁止作业抢占,工件一旦开始加工,则必须持续进行,不得中途暂停或插入其他工件;
- (3) 在时间零点时,所有工件均可开始加工,机器均可用;
- (4) 每个工件的各个工序在不同机器上的作业时间事先已知,机器的准备时间与操作顺序无关,并且已经包含在作业时间内;
- (5) 工件按照流水线的工艺约束进行加工,在完成前一个工序后才能在下一个工序上加工,传输时间忽略不计;
- (6) 对于每个工件的每一道工序,仅能从并行机中选取一台机器进行加工;
- (7) 每台机器一次只能加工一个工件。

2.2 符号定义

排产问题数学模型所使用的符号及其描述如表 1 所列。

表 1 模型中符号定义与描述

Table 1 Symbol definition and description in the model

符号	描述
n	工件数量
m	工序数量
M_j	第 j 道工序的可用机器集
T_{ijk}	工件 i 在工序 j 的机器 k 上的作业时间
S_{ijk}	工件 i 在工序 j 的机器 k 上的起始作业时刻
E_{ijk}	工件 i 在工序 j 的机器 k 上的结束作业时刻
C_{jk}	工序 j 中机器 k 的加工工件数量
P_{jk}	在工序 j 中机器 k 上加工的第 P 个工件
x_{ijk}	当工件 i 在工序 j 的机器 k 上进行加工时 $x_{ijk} = 1$, 否则 $x_{ijk} = 0$

2.3 数学模型

面对有限的资源,如何在有限时间内完成更多的生产指标以实现最大的经济效益,是大多数生产制造型企业的目标之一。因此,以完工时间最小为排产优化目标,即 *makespan*。目标函数为:

$$f = \min(\sum_{k=1}^{M_j} E_{ijk}); i = 1, 2, \dots, n; j = 1, 2, \dots, m; k \in M_j \quad (1)$$

根据约束条件,每个工件必须分配在各个工序的某一台

机器上,即同一工序中所有并行机上加工的工件总数为全部的工件数之和 n ,如式(2)、式(3)所示。工件下一工序加工必须在上一工序流程结束后才能开始,即工件加工的工序约束,如式(4)所示。工件的完工时刻表示为工件开始加工的时刻加上工件在已选择的机器上的加工用时,如式(5)所示。在同一时间内,每台机器只能加工一个工件,如式(6)所示。式(7)为决策变量,仅存在 0 和 1 两个值。

$$\sum_{k=1}^{M_j} x_{ijk} = 1 \quad (2)$$

$$\sum_{k=1}^{M_j} C_{jk} = n \quad (3)$$

$$E_{ijk} \leq S_{i(j+1)t} \quad (4)$$

$$E_{ijk} = S_{ijk} + T_{ijk} \quad (5)$$

$$E_{P_{ik}jk} \leq S_{(P+1)jk} \quad (6)$$

$$x_{ijk} = \begin{cases} 1, & \text{工件 } i \text{ 在工序 } j \text{ 的机器 } k \text{ 上加工} \\ 0, & \text{否则} \end{cases} \quad (7)$$

其中, $i=1, 2, \dots, n; j=1, 2, \dots, m; k \in M_j; t \in M_{j+1}; P=1, 2, \dots, C_{jk}-1$ 。

3 混合禁忌搜索的改进遗传算法

结合上述排产问题的特点和数学模型,排产方案既包括工件生产顺序的赋值,还包括选择合适的并行机进行加工,因此更具有挑战性。本文设计了一个混合禁忌搜索遗传优化算法(Improved Genetic Algorithm with Tabu Search, IGA-TS)进行求解,优化排产。

3.1 算法总体流程

将局部搜索算法混合到遗传算法中的设计方法有很多种,本文将禁忌搜索引入到遗传算子后,具体的算法说明将在下文进行阐述。混合禁忌搜索的遗传优化算法的详细流程如图 2 所示。

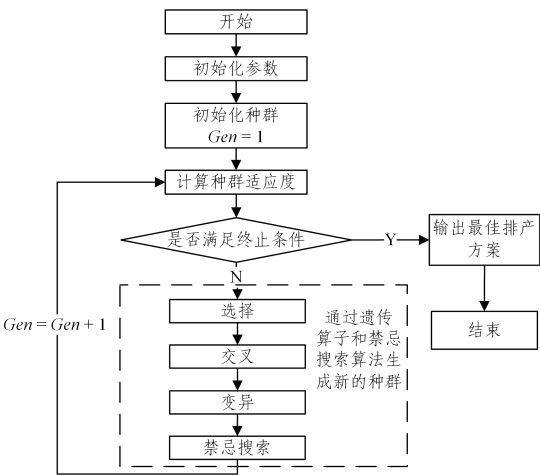


图 2 混合禁忌搜索的遗传优化算法流程图

Fig. 2 Flow chart of improved genetic algorithm with tabu search

具体步骤如下:

- 步骤 1 设置遗传算法的参数。
- 步骤 2 初始化种群,设 $Gen=1, Gen$ 为当前代。
- 步骤 3 根据优化目标对种群中每个个体计算适应度。
- 步骤 4 是否满足终止条件? 若满足,执行步骤 7; 否则,

执行步骤 5。

步骤 5 开始生成新种群。

步骤 5.1 应用遗传算子产生新种群,其中包括选择、交叉和突变等。

步骤 5.2 应用禁忌搜索增强部分个体适应度。

步骤 6 设置 $Gen=Gen+1$, 执行步骤 3。

步骤 7 输出最佳排产方案。

3.2 编码与解码

在遗传算法中,个体的编码方式非常重要,它关系到解空间的优劣、后续阶段的计算难度和进化的效果,每个染色体对应一种排产方案。本阶段的主要任务为初始化算法参数,对排产计划进行编码,产生初代染色体种群。其中, $maxGen$ 表示总迭代次数, $PopSize$ 表示种群规模, Pc 表示交叉概率, Pm 表示变异概率。

鉴于流水生产线的特点,后续阶段工件的加工顺序受上一阶段影响较大,本文算法采用基于工件排列的编码方法。取所有待加工工件的一种排序方式作为一个染色体,染色体长度 $R_i = n (i=1, 2, \dots, PopSize)$, 工件在染色体中的位置则表示工件首道工序的加工顺序。后续阶段根据前一阶段工件加工完成时间,采用非递增顺序的排列方式进行编码。例如,对于有 4 道加工工序的混合流水线 ($m=4$), 生产 5 个工件 ($n=5$), 以数字 1, 2, 3, 4, 5 指代不同的工件。假设 (4, 2, 5, 1, 3) 为生成的初始染色体编码, 即第一阶段加工工件顺序, 工件 4 先加工, 紧接着是工件 2, 其次是工件 5, 再是工件 1, 最后是工件 3。若上述 5 个工件进入第二阶段的顺序为 2, 4, 1, 3, 5, 则第二阶段工序的染色体编码为 (2, 4, 1, 3, 5)。后续阶段同理, 待 4 个阶段全部完成后, 可得到一种排产方法。图 3 给出了一个编码流程示例。

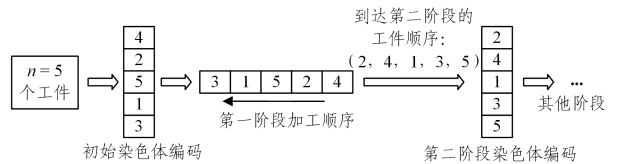


图 3 染色体编码示例

Fig. 3 Example of chromosome coding

对于大规模的工件排产问题,上述编码方式生成的染色体规模小,操作简单。依据此编码,采用一定的规则进行解码,对于待处理的工件各阶段,选择合适的机器,可以得到排产方案在该编码方式下的最大完工时间。

对于每个染色体,解码算法的具体步骤如下:

步骤 1 对于所有工件在每道工序的可用机器集合,分别计算机器在工序上的起始时间、结束时间、作业时长以及该机器的负荷。

步骤 2 机器选择的优先级设定如下:第一优先级为前工序结束时间,即机器的前工序结束时间越短,优先级越高;第二优先级为机器的工作负荷,即机器的工作负荷越轻,优先级越高;当机器的第一、第二优先级均相同时,则随机选取。

步骤 3 检查所有工件的所有工序是否都已完成机器的选择。若全部完成,即产生了排产方案,其中包括工件在每个

阶段的加工顺序和异步并行机的选择情况,输出针对该染色体的解码结果、排产方案时间表和适应度大小。

上述编码和解码方法,针对异步并行机混合流水车间排产的两个关键问题分别提供了合适的解决策略,既能减小大规模场景下排产方案染色体大小对后续选择、交叉、变异计算量的影响,又能根据运行流程中并行机的实际情况选择合适的机器进行加工,保证每个染色体编码都是可执行的排产方案,避免对不可行解的计算和迭代。

3.3 遗传算子

采用良好的遗传算子,在种群中高效地产生优秀个体,扩大解空间,是遗传算法中非常重要的环节。遗传算子一般可分为3类:选择、交叉和突变^[16]。由于本文算法中,机器选择主要在解码环节进行,因此遗传算子主要作用于待加工工件生产顺序的安排。

3.3.1 选择

在本文算法中,根据排产优化目标,即最大完工时间,计算种群中个体的适应度,选取 $PopSize$ 个染色体进入交叉阶段。本文算法采用两种选择算子。第一个是轮盘赌选择法,根据式(8)计算出每个个体被选中的概率,其中选中概率在 $[0,1]$ 之间,生成的随机数落在哪个个体的区间里,则选择该个体进入下一代。

$$p(x_i) = \frac{f(x_i)}{\sum_{j=1}^n f(x_j)} \quad (8)$$

另一个是锦标赛选择制度,即在种群中随机选择一定数量 N 的个体(锦标赛规模 N 一般在 $2 \sim 7$ 之间,本文算法根据对比实验结果,选取 $N=2$,效果更好),将 N 个个体中适应度最好的个体保留至下一代。锦标赛选择法也可以通过改变比赛的规模来调节选择的性能。两种选择算子分别操作现有种群一半大小的染色体,两种方式结合能有效平衡随机选择和保留原始种群内优秀个体的矛盾,优化算法收敛效果。

3.3.2 交叉

在遗传算法中,种群通过交叉两个父代染色体的部分区域生成子染色体。当生成的子染色体拥有比父代染色体更优异的适应度时,则表明种群实现了进化。本文算法使用两种交叉算子,在算法中任意一对父代染色体随机选择一种交叉算子生成子代染色体。

(1) 工件优先交叉算子(Job-Base Crossover, JBX)。具体步骤如下(父代个体记为 P_1 和 P_2 ,子代个体记为 O_1 和 O_2):

步骤1 将工件集 $n = \{n_1, n_2, \dots, n_n\}$ 随机分成两个子集 Job-set1 和 Job-set2。

步骤2 将 P_1 个体中除去 Job-set1 子集的工件序按照原位置复制进个体 O_1 中;同理,将 P_2 个体中除去 Job-set1 子集的工件序按照原位置复制进个体 O_2 中。

步骤3 把 P_2 个体中属于子集 Job-set1 中的工件按原顺序插入 O_1 个体中的空缺位置,即得到交叉后的新个体 O_1 。同理,可得新个体 O_2 。

图4为两个父代染色体 $(2, 1, 4, 5, 3)$ 和 $(3, 2, 5, 1, 4)$ 以 Job-set1 = $\{2, 3\}$ 和 Job-set2 = $\{1, 4, 5\}$ 为两个工件子集,按照 JBX 进行交叉操作的实际流程示意图。

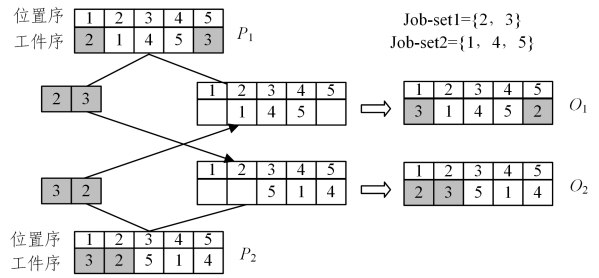


图4 JBX交叉算子操作

Fig. 4 Job-base crossover operation

(2) 位置优先交叉算子(Precedence Operation Crossover, POX)。具体步骤如下(父代个体记为 P_1 和 P_2 ,子代个体记为 O_1 和 O_2):

步骤1 将工件集 $n = \{n_1, n_2, \dots, n_n\}$ 基于所在位置随机分成两个子集 Operation-set1 和 Operation-set2。

步骤2 将 P_1 个体中与 P_2 个体子集 Operation-set1 不重叠的工件序按照原位置复制进个体 O_1 中;同理,将 P_2 个体中与 P_1 个体子集 Operation-set1 不重叠的工件序按照原位置复制进个体 O_2 中。

步骤3 将 P_2 个体子集 Operation-set1 中的工件序插入 O_1 个体中的相应位置,即得到交叉后的新个体 O_1 ,且不存在重复的工件序。同理,可得新个体 O_2 。

图5为两个父代染色体 $(2, 1, 4, 5, 3)$ 和 $(3, 2, 5, 1, 4)$ 以 Operation-set1 = $\{2, 3\}$ 和 Operation-set2 = $\{1, 4, 5\}$ 为两个工件子集,按照 POX 进行交叉操作的实际流程示意图。

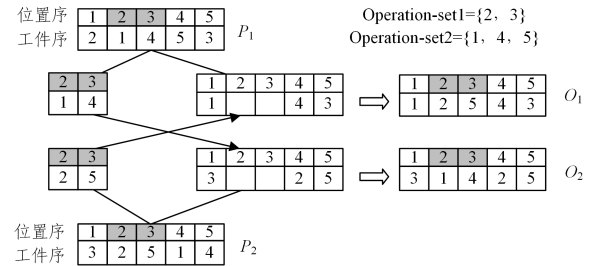


图5 POX交叉算子操作

Fig. 5 Precedence operation crossover operation

位置优先交叉算子以染色体上的位置顺序为基础,需要考虑各染色体中不存在重复工件号的约束条件,对实现逻辑和计算有一定的要求,但是对交叉后子代染色体的改动较为明显。两种交叉算子相辅相成,既能保证对不同规模染色体的交叉操作效果,也能均衡算法的计算难度。

3.3.3 变异

根据变异概率,选取染色体并使用变异算子操作得到新的染色体个体。本文算法使用随机交换突变算子,这种突变算子对染色体产生的扰动较大,随机性更强,能够更好地跳出局部最优,扩大解空间。

变异算子的具体步骤如下:

步骤1 在父代染色体中随机选择 $r (1 < r < n)$ 个位置。

步骤2 打乱交换 r 个位置中的工件序号,生成子代染色体。

3.4 禁忌搜索算法

禁忌搜索算法是一种模拟人类智能的优化算法,通过禁

忌表限制搜索区域,避免迂回搜索。禁忌搜索是目前解决调度问题最有效的局部搜索之一。引入禁忌搜索,可以针对前期遗传算法生成的排产方案进行局部调优,扩大解空间,定向改进排产结果,加快算法收敛速度。算法主要由4个元素组成,包括搜索算子、禁忌列表、藐视准则和终止准则。搜索算子是一种通过对给定解引入微小扰动而产生邻域解的机制^[17]。在禁忌搜索算法中,从现有解到新解的过渡受到这种结构的显著影响,不合理的搜索算子会导致算法搜索效率明显降低。在设计搜索算子时,还需要排除冗余和不可行的改变,在不影响解质量的前提下,选择合适的搜索算子。目前的搜索算子主要作用于解的头部或尾部。

本算法采用了两种搜索算子,即N7算子^[17]和k-insertion算子^[18],它们已被证明在混合流水车间排产问题(HFSP)中是有效的^[19]。本文经过多次实验对比,选择10%的个体进行禁忌搜索定向优化。由于两个算子分别改变了同一加工阶段内、同一机器工件加工的顺序以及工件加工机器的选择,因此所选择的染色体个体在进入搜索阶段之前已经使用解码方法确定了当前的机器选择情况和适应度的大小。同时,N7算子和k-insertion算子都设计了特定的约束,保证了所有生成解的可行性,能提高候选解的搜索效率。

禁忌搜索流程图如图6所示。

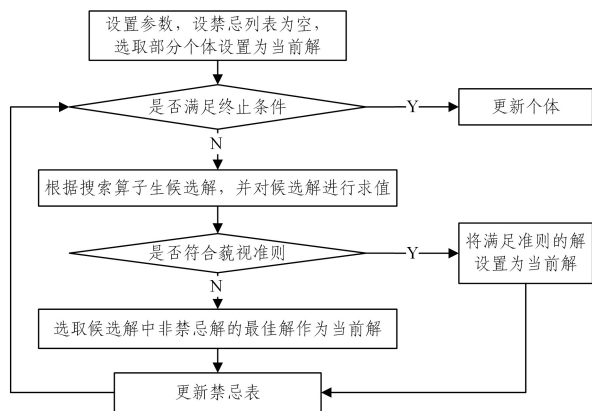


图6 禁忌搜索流程图

Fig. 6 Tabu search flow chart

步骤1 设定禁忌搜索参数,初始化禁忌列表为空,选取部分个体设为初始解。

步骤2 是否满足终止条件?若满足,执行步骤8;否则执行步骤3。

步骤3 通过搜索算子生成新的候选解,并对候选解进行解码计算和适应度求值。

步骤4 是否满足藐视标准?若满足,执行步骤6;否则执行步骤5。

步骤5 在候选解中选择非禁忌解的最佳解设为当前解,执行步骤7。

步骤6 将满足藐视标准的解设为当前解。

步骤7 更新禁忌列表,执行步骤2。

步骤8 输出最优方案。

3.5 终止条件

在本算法中,当算法迭代代数达到最大值($maxGen$)或达

到允许的最大滞留步长($maxStayGen$)而没有改进时,算法终止;对禁忌搜索环节,当迭代次数达到最大($maxTSIterSize$)时,对该个体的禁忌搜索终止。

4 实验及结果

4.1 实验数据及环境

为了评估所提IGA-TS算法的有效性,借鉴两个标准算例的设计思想:Carlier等^[20]提出的77个HFSP同速并行机中c和d两种较复杂类型的标准算例;Liao等^[21]提出的10个包含30个工件的大规模算例,规定实验算例包含5至10道工序,不同类型工序存在不同的生产瓶颈。同时,结合文献^[22]中调研的A企业异步并行机排产车间问题的实际数据,同种工件在同一工序不同机器上的加工时间分布从4min到17min不等;而在工件数量增大的情况下,涉及到机器调整时间,工件的加工时间会略有增加。因此,本文针对异步并行机的混合流水车间排产问题,随机生成30个中规模算例(c,d类型)和10个大规模算例(e类型)分别进行实验分析。每个算例由工件数量、工序数量和各工序上的可用机器配置来定义。机器配置的结构用字母c,d,e表示,各类型的说明如下。

c:中间一道工序有两台可选机器,其他阶段为3台(存在一个瓶颈阶段)。

d:各阶段的工序均有3台可选机器(无瓶颈阶段)。

e:各个工序的机器数量在(3,5)间随机生成。

中规模算例包含10个工件5道工序、10个工件10道工序和15个工件5道工序3种情况,大规模算例包含30个工件5道工序。其中,中规模算例中不同工序的加工时间在[3,20]之间随机生成,大规模算例中不同工序的加工时间在[20,50]之间随机生成。

例如,一个有10个工件和5道工序的算例用j10c5d2表示,其中字母j表示工件数,字母c表示工序数,字母d表示机器配置类型,最后一个数字2表示该配置下的第二个算例。

本算法以PyCharm 2023为开发环境,python3.8.1为编程语言,实验的运行环境为Intel(R) Core(TM) i7-12700 @ 2.10GHz,16GB运行内存,Windows11 64位操作系统。

4.2 算法参数选择

混合禁忌搜索的遗传优化算法需要提前设定种群大小($PopSize$)、最大迭代次数($maxGen$)、最大滞留代数($maxStayGen$)、交叉概率(Pc)、突变概率(Pm)、禁忌搜索表长度($TSlist$)和禁忌搜索最大迭代次数($maxTSIterSize$)7个参数。经过对类似混合流水车间问题规模的文献^[23-24]的研究,确定 $PopSize$ 、 $maxGen$ 和 $maxStayGen$ 这3个参数。种群规模设定过大会增加计算量,过小则无法提供足够的采样点;最大迭代次数过多会导致运行时间过长,过少会导致时间不足而无法出现最优解。对于交叉概率、突变概率、禁忌搜索表长度等对算法搜索和优化性能有较大影响的参数,本文使用正交实验确定参数取值,参数水平如表2所列。实验数据选取中规模算例j15c5c4和大规模算例j30c5e2,每种参数组合对每个算例独立测试5次,取平均值消除随机性。实验结果如表3所列。图7为3个参数不同水平下对应的最大完工时间平均值的变化趋势图。

表2 参数水平

Table 2 Levels of parameters

参数	水平值		
	1	2	3
P_c	0.9	0.8	0.7
P_m	0.1	0.2	0.3
$TSlist$	6	9	12

表3 正交实验结果

Table 3 Orthogonal experimental results

序号	P_c	P_m	$TSlist$	$Time_Avg$
1	1	1	1	705.4
2	1	2	2	701.0
3	1	3	3	704.7
4	2	1	2	704.4
5	2	2	3	703.6
6	2	3	1	703.4
7	3	1	3	702.6
8	3	2	1	701.8
9	3	3	2	699.4

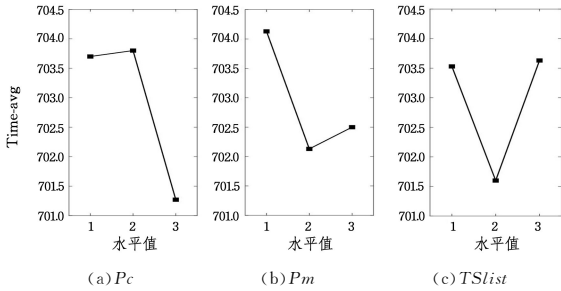


图7 最大完工时间平均值随水平变化的趋势图

Fig. 7 Trend of the average maximum completion time with the levels change

根据表3和图7,当参数组合为 $[3, 2, 2]$ 时,算法表现最好,算例最大完工时间的平均值最小,因此本文采用 $P_c = 0.7, P_m = 0.2, TSlist = 9$ 进行后续实验。表4对正交实验结果进行了极差分析。由表4可知, P_c 参数对算法影响最大,其次是 $TSlist, P_m$ 。此外, $TSlist$ 和 P_m 的极差结果相仿,所以两个参数对算法的影响相似,也可以证明禁忌搜索算法对于优化遗传算法的有效性。

表4 极差分析

Table 4 Range analysis

水平值	P_c	P_m	$TSlist$
1	703.70	704.13	703.53
2	703.80	702.13	701.60
3	701.27	702.50	703.63
range	2.10	2.00	2.03
level	1.00	3.00	2.00

在IGA-TS算法进化的早期,由于遗传算法不能为禁忌搜索提供较好的初始个体,禁忌搜索找到更优解的可能性很小,因此将禁忌搜索的最大迭代次数设定为:

$$\max TSIterSize = 800 \times (Gen / \max Gen) \quad (9)$$

最大迭代次数在算法进化过程中自适应调整,这样可以很好地平衡算法的开发和搜索,节省计算时间。

IGA-TS算法相关参数设定如表5所列。

表5 算法相关参数

Table 5 Algorithm related parameters

参数	取值
种群大小	100
最大迭代次数	200
最大滞留代数	50
交叉概率	0.7
突变概率	0.2
禁忌搜索最大迭代次数	$800 \times (Gen / \max Gen)$
禁忌表长度	9

4.3 中规模算例求解

本文采用迭代贪婪算法(Iterative Greed Algorithm, IG)、标准遗传算法(Genetic Algorithm, GA)、粒子群算法(Particle Swarm Optimization, PSO)、改进遗传算法(Hybrid Genetic Algorithm and NEH, HGA)^[25]与混合禁忌搜索遗传优化算法(IGA-TS)进行了实验结果的对比分析。

30个中规模算例c与d的实验结果如表6、表7所列。

表6 算例实验结果(c类型)

Table 6 Experimental results(type c)

算例	IG	GA	PSO	HGA	IGA-TS
j10c5c1	72	59	59	59	59
j10c5c2	81	63	63	63	63
j10c5c3	77	71	71	71	71
j10c5c4	75	69	71	64	64
j10c5c5	75	62	68	62	62
j10c5c6	76	66	63	66	63
j15c5c1	86	82	79	79	77
j15c5c2	83	74	74	74	74
j15c5c3	88	76	79	77	76
j15c5c4	87	81	80	75	75
j15c5c5	79	76	76	76	76
j15c5c6	76	69	69	69	69
j10c10c1	123	106	113	105	105
j10c10c2	116	98	98	101	98
j10c10c3	121	103	103	103	103
j10c10c4	108	99	98	98	98
j10c10c5	132	112	115	109	109
j10c10c6	125	113	108	108	103

表7 算例实验结果(d类型)

Table 7 Experimental results(type d)

算例	IG	GA	PSO	HGA	IGA-TS
j10c5d1	75	70	70	70	70
j10c5d2	88	86	86	82	82
j10c5d3	79	76	74	76	74
j10c5d4	92	86	86	87	86
j10c5d5	77	67	69	67	67
j10c5d6	86	71	70	71	70
j15c5d1	99	87	87	87	87
j15c5d2	97	93	93	89	89
j15c5d3	119	99	104	99	98
j15c5d4	96	84	84	84	84
j15c5d5	121	106	103	99	94
j15c5d6	101	90	89	89	89

分析表6、表7可知,IGA-TS算法均在30个中规模算例中找到了最优解,而HGA算法找到了20个算例中的最优解,PSO算法找到了17个算例中的最优解,GA算法找到

15 个算例中的最优解。

GA 算法、PSO 算法、HGA 算法和 IGA-TS 算法在中规模算例中,对于 IG 算法的平均相对改进百分比如表 8 所列。

表 8 算法相对改进百分比(c,d 类型)

Table 8 Relative percent gap of algorithms(type c,d)

算法	RG/%
GA	11.04
PSO	10.77
HGA	12.23
IGA-TS	13.10

其中,相对改进百分比(Relative Percent Gap, RG)的计算式为:

$$RG(r_1) = \frac{r_1 - r_2}{r_2} \times 100\% \quad (10)$$

其中, r_1 和 r_2 分别为待对比算法和 IG 算法的最大完工时间。

由此可知,IGA-TS 算法的平均相对改进百分比最佳,表现稳定,在中规模算例上的表现均优于其他对比算法。

为了进一步验证 IGA-TS 算法对于优化遗传算法的收敛效果,针对 IGA-TS 算法和 HGA 算法在中规模算例中生成相同最优解的 20 个算例,对比算法单次迭代时间和生成最优解所需的迭代次数。表 9 列出了两种算法在 20 个算例中的对比结果,其中 T_1 为迭代一次所需的平均时间; C_{iter} 为算法生成最优解所需的迭代次数; $T_{optimal}$ 为首次达到最优结果所需的时间。

表 9 最优解迭代次数和时间

Table 9 Iteration times and time of optimal solutions

算例	HGA			IGA-TS		
	T_1	C_{iter}	$T_{optimal}$	T_1	C_{iter}	$T_{optimal}$
j10c5c1	0.34	27	9.25	0.45	23	10.28
j10c5c2	0.36	10	3.56	0.54	7	3.76
j10c5c3	0.38	20	7.60	0.54	12	6.52
j10c5c4	0.36	29	10.43	0.51	7	3.55
j10c5c5	0.36	57	20.56	0.52	5	2.61
j15c5e2	0.39	65	25.56	0.48	51	24.61
j15c5c4	0.49	91	44.89	0.61	70	42.88
j15c5c5	0.58	95	55.08	0.63	43	27.05
j15c5c6	0.47	72	34.05	0.69	54	37.23
j10c10c1	0.84	28	23.39	0.95	14	13.25
j10c10c3	0.80	14	11.18	0.95	17	16.22
j10c10c4	0.82	32	26.36	0.92	33	30.24
j10c10c5	0.70	9	6.29	1.00	3	2.99
j10c5d1	0.44	44	19.34	0.56	42	23.60
j10c5d2	0.45	53	23.88	0.57	31	17.62
j10c5d5	0.45	20	8.91	0.57	13	7.42
j15c5d1	0.62	60	37.04	0.74	27	20.04
j15c5d2	0.63	26	16.28	0.77	36	27.68
j15c5d4	0.59	55	32.65	0.77	55	42.11
j15c5d6	0.58	67	38.78	0.78	44	34.21

分析表 9 可以看出,IGA-TS 算法在 16 个算例中产生同样解的迭代次数明显少于 HGA 算法,所需迭代次数减少了 25.72%。IGA-TS 算法由于设计更为复杂,每一次迭代所需的计算时间更长,但对比首次达到最优结果所需的时间,IGA-TS 算法平均所需迭代时间更短,算法首次达到最优结

果的运行时间减少了 10.79%。面对同样规模的问题,在达到相同解的情况下,IGA-TS 算法能明显提高收敛速度,减少迭代次数,缩短算法的运行时间。

图 8 是算例 j15c5d1 的 IGA-TS 算法和 HGA 算法迭代收敛曲线对比图。

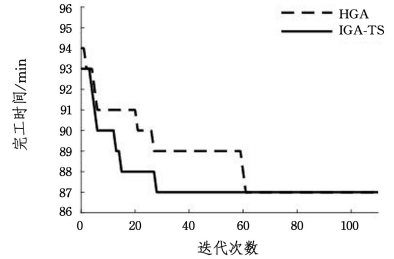


图 8 算例 j15c5d1 的算法迭代变化趋势曲线对比图

Fig. 8 Instance j15c5d1 iteration trend curve comparison

4.4 大规模算例求解

为进一步验证 IGA-TS 算法求解大规模异步并行机混合流水车间排产问题的有效性,采用 10 个大规模算例(e 类型)进行求解,实验结果如表 10 所列。

表 10 算例实验结果(e 类型)

Table 10 Experimental results(type e)

算例	IG	GA	PSO	HGA	IGA-TS
j30c5e1	671	641	636	627	616
j30c5e2	668	638	631	623	617
j30c5e3	679	644	625	627	625
j30c5e4	664	638	637	638	634
j30c5e5	687	633	646	624	620
j30c5e6	701	622	629	589	590
j30c5e8	682	639	648	641	639
j30c5e9	640	605	611	594	593
j30c5e10	653	597	609	597	597

分析表 10 可以得出,IGA-TS 算法在 10 个大规模算例中求出了 9 个算例的最优解,而 GA 算法、PSO 算法、HGA 算法分别求出了 2,1,2 个算例的目前最好解。因此,针对工件数量大、各工序可选机器数量多的复杂排产场景,IGA-TS 算法生成的解质量明显优于其他算法。

GA 算法、PSO 算法、HGA 算法和 IGA-TS 算法在中规模算例中,对于 IG 算法的平均相对改进百分比如表 11 所列。

表 11 算法相对改进百分比(e 类型)

Table 11 Relative percent gap of algorithms(type e)

算法	RG/%
GA	6.34
PSO	6.02
HGA	7.23
IGA-TS	8.31

分析表 11 可知,IGA-TS 算法的改进效果最佳,相对于 IG 算法的平均改进百分比为 8.31%。

为了验证算例规模对算法执行时间的影响,设计了 10 个包含 5 道工序 45 个工件、10 个包含 5 道工序 60 个工件的大规模算例进行实验,计算各个算例首次达到最优结果时算法执行时间的平均值。图 9 为算法平均执行时间随工件规模的变化趋势图。

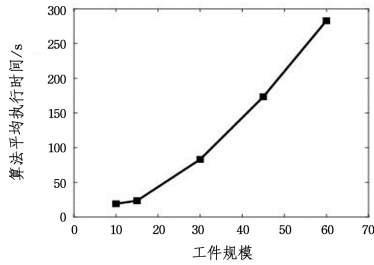


图9 算法平均执行时间随工件规模的变化趋势图

Fig. 9 Variation trend of the average execution time of the algorithm with the size of the workpiece

分析图9可知,算法平均执行时间随着工件规模的增大而增大,二者呈线性关系。因此,本算法的性能不会因为算例规模的增大而明显劣化,其在更大规模的算例中具有可行性。

图10为算例j30c5e1的排产方案甘特图,矩形框内(n, m)的含义为“第n个工件的第m道工序”。排产方案甘特图,可以直观高效地指导排产流程,安排生产计划。

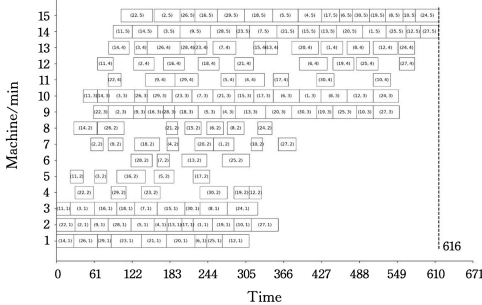


图10 算例j30c5e1的调度方案甘特图

Fig10 Instance j30c5e1 scheduling scheme Gantt chart

4.5 加工车间案例应用

以某一机加车间为例,现需要加工9个工件,每个工件都需要经过3道工序加工处理,分别是车、铣、磨,车间内有2台车床、3台铣床、2台磨床,且不同工件在机床上的加工时间不同,具体加工情况如表12所列。

表12 加工时间明细

Table 12 Processing time list

工件	车1	车2	铣1	铣2	铣3	磨1	磨2
1	16	15	12	15	14	14	16
2	14	12	15	13	16	14	15
3	13	15	13	15	19	13	15
4	14	15	12	14	15	16	16
5	15	16	17	14	17	13	15
6	15	14	14	17	15	16	17
7	15	13	15	13	18	17	13
8	12	14	16	14	19	18	16
9	16	14	15	12	18	14	17

将3个对比算法和IGA-TS算法应用于车间案例进行对比,得到的结果如表13所列。

表13 各算法排产结果

Table 13 Scheduling results of each algorithm

算法	最优解用时/min	最优解迭代次数
GA	99	35
PSO	99	38
HGA	97	21
IGA-TS	95	7

通过比较分析可知,IGA-TS算法在实际应用案例中产生了比其他算法更优的排产方案,且达到最优方案需要的迭代次数更少,算法搜索效果更好。图11为IGA-TS算法的排产方案甘特图。

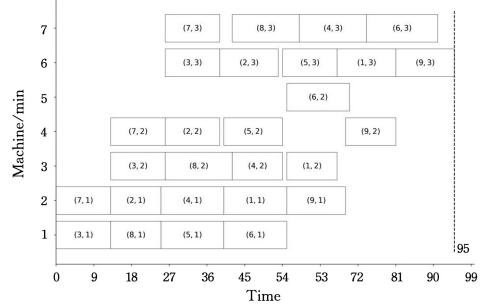


图11 加工车间案例排产方案甘特图

Fig. 11 Gantt chart of workshop scheduling scheme

根据上述实验结果,IGA-TS算法对求解异步并行机混合流水车间中的中规模和大规模算例均具有良好的效果。IGA-TS算法结合传统遗传算法的全局搜索、群体进化的优势,又针对个体进行禁忌搜索优化,既扩大了搜索范围,也能在保证解质量的同时有效提高收敛速率。在计算中规模算例的最大完工时间时,IGA-TS算法所产生的解明显优于IG算法、GA算法、PSO算法和HGA算法,且能在更少的迭代次数和运行时间内达到与HGA算法相同的解。面对复杂度更高的大规模算例求解,IGA-TS算法同样展现出优势,得出10个大规模算例中9个算例的目前最优解。IGA-TS算法生成的排产结果的完工时间相较于IG算法平均缩短了10.71%,达到最优解所需的迭代次数相较于HGA减少了25.72%,算法达到最优解所需的运行时间缩短了10.79%。在车间案例实际应用中,所提算法不仅优化了排产方案的最大完工时间,也提高了搜索效率,减少了迭代次数。

结束语 异步并行机混流车间生产排产问题是众多制造企业的主要生产场景,且该排产模型也能迁移应用到许多新兴领域,具有广泛的应用前景。本文首先对异步混合流水生产场景进行建模,根据工件数量多、各阶段并行机不同速的特点,设计排产方案的染色体结构、编码和解码流程,所生成的染色体结构简单,对后续环节构成的计算压力小,根据解码方法还能择优选择加工机器。然后,使用两种选择算子、两种交叉算子和一种变异算子生成子代染色体,扩大种群规模,再通过禁忌搜索算法优化种群中随机10%的个体,定向提高种群适应度以及解的质量。经过多次迭代,在有限的时间内得到目前最优的排产计划。最后,使用30个中规模复杂算例、10个大规模算例和一个加工车间案例数据,将迭代贪婪算法、传统遗传算法、粒子群算法、混合遗传算法与本文所提算法进行对比。实验结果表明,混合禁忌搜索遗传优化算法在中规模和大规模的混合流水车间排产问题中,收敛速度明显提高,排产方案的优化效果明显,能在有限资源中缩短生产时间,为企业带来更大的效益。

在求解过程中也发现了一些不足,可以作为后续的研究方向:(1)没有考虑多种优化目标的需求,未来可以进一步优化算法或调整参数,增强算法多目标优化的能力;(2)在实际

生产中可能遇到紧急订单插入的情况,后续可以在本文算法的基础上结合机器学习算法提高算法响应的实时性。

参 考 文 献

- [1] LI Y L, LI X T, GAO L. A survey of hybrid flow shop scheduling problems [J]. *China Mechanical Engineering*, 2020, 31(23): 2798-2813, 2828.
- [2] RAUF M, GUAN Z, SARFRAZ S, et al. A smart algorithm for multi-criteria optimization of model sequencing problem in assembly lines[J]. *Robotics and Computer-Integrated Manufacturing*, 2020, 61: 101844.
- [3] MOSADEGH H, GHOMI S M T F, SÜER G A. Stochastic mixed-model assembly line sequencing problem: Mathematical modeling and Q-learning based simulated annealing hyper-heuristics[J]. *European Journal of Operational Research*, 2020, 282(2): 530-544.
- [4] ZHOU Q Y, XIAO M S, ZHANG L X, et al. Intelligent optimization technology for production scheduling under multiple constraints [J]. *Computer Science*, 2021, 48(3): 239-245.
- [5] LV Y Y, FAN K, QU H, et al. Multi-objective particle swarm optimization for hybrid multiprocessor task shop scheduling problem [J]. *Microcomputer System*, 2022, 43(1): 218-224.
- [6] LI Y, LI X, GAO L, et al. A discrete artificial bee colony algorithm for distributed hybrid flowshop scheduling problem with sequence-dependent setup times [J]. *International Journal of Production Research*, 2021, 59(13): 3880-3899.
- [7] LU C, GAOL, PAN Q, et al. A multi-objective cellular grey wolf optimizer for hybrid flowshop scheduling problem considering noise pollution[J]. *Applied Soft Computing*, 2019, 75: 728-749.
- [8] DONG J, YE C M. Interval number reentrant hybrid flow shop scheduling and pre-maintenance collaborative optimization [J]. *Control and Decision-making*, 2021, 36(11): 2599-2608.
- [9] ZHOU Y Q, WANG C Y, LI Y L, et al. Improved fruit fly algorithm for hybrid flow shop scheduling problem [J]. *Control Theory and Application*, 2023, 40(4): 597-606.
- [10] ZHANG B, XU L, ZHANG J. A multi-objective cellular genetic algorithm for energy-oriented balancing and sequencing problem of mixed-model assembly line [J]. *Journal of Cleaner Production*, 2020, 244: 118845.
- [11] DEFERSHA F M, MOHEBALIZADEHGASHTI F. Simultaneous balancing, sequencing, and workstation planning for a mixed model manual assembly line using hybrid genetic algorithm[J]. *Computers & Industrial Engineering*, 2018, 119: 370-387.
- [12] YUAN S, LI T K, WANG B L. Improved migratory bird optimization algorithm for group scheduling problem of hybrid flow shop with unrelated parallel machines [J]. *Computer Integrated Manufacturing System*, 2022, 28(12): 3912-3922.
- [13] LOPES T C, SIKORA C G S, MICHELS A S, et al. An iterative decomposition for asynchronous mixed-model assembly lines: combining balancing, sequencing, and buffer allocation[J]. *International Journal of Production Research*, 2020, 58(2): 615-630.
- [14] ABREU L R, PRATA B A. A hybrid genetic algorithm for solving the unrelated parallel machine scheduling problem with sequence dependent setup times[J]. *IEEE Latin America Transactions*, 2018, 16(6): 1715-1722.
- [15] XUAN H, ZHENG Q, LI B, et al. A novel genetic simulated annealing algorithm for no-wait hybrid flowshop problem with unrelated parallel machines[J]. *ISIJ International*, 2021, 61(1): 258-268.
- [16] OMARA F A, ARAFA M M. Genetic algorithms for task scheduling problem[J]. *Journal of Parallel and Distributed computing*, 2010, 70(1): 13-22.
- [17] 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 [J]. *Computers & Operations Research*, 2007, 34(11): 3229-3242.
- [18] MASTROLILLI M, GAMBARDELLA L M. Effective neighbourhood functions for the flexible job shop problem[J]. *Journal of scheduling*, 2000, 3(1): 3-20.
- [19] ZHOU H, LIU H, LV C, et al. A Path Relinking with Tabu Search Algorithm for Solving Hybrid Flow Shop Scheduling Problem Considering Multiple Critical Paths [J]. *Computer & Operations Research*, 2024, 170: 106783.
- [20] CARLIER J, NERON E. An exact method for solving the multiprocessor flow-shop [J]. *RAIRO-Operations Research-Recherche Opérationnelle*, 2000, 34(1): 1-25.
- [21] LIAO C J, TJANDRADAJA E, CHUNG T P. An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem [J]. *Applied Soft Computing*, 2012, 12(6): 1755-1764.
- [22] LIU C. Application of genetic algorithm in parallel multi-locomotive scheduling problem of enterprise A [D]. Shanghai: Donghua University, 2022.
- [23] YU C, SEMERARO Q, MATTA A. A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility [J]. *Computers & Operations Research*, 2018, 100: 211-229.
- [24] SUN L, SHI W, WANG J, et al. Research on production scheduling technology in knitting workshop based on improved genetic algorithm[J]. *Applied Sciences*, 2023, 13(9): 5701.
- [25] FAN J, LI Y, XIE J, et al. A hybrid evolutionary algorithm using two solution representations for hybrid flow-shop scheduling problem[J]. *IEEE Transactions on Cybernetics*, 2021, 53(3): 1752-1764.



WANG Sitong, born in 1999, postgraduate. Her main research interests include industrial big data analysis and so on.



LIN Rongheng, born in 1981, Ph.D, professor, Ph.D supervisor. His main research interests include industry big data analysis and decision-making, and industrial big data analysis.