

文章编号: 1008-1542(2023)02-0152-13

考虑工人疲劳和学习效应的装备制造 车间动态调度

郭超^{1,2}, 郝东辉³, 郭鹏^{1,4}

(1. 轨道交通运维技术与装备四川省重点实验室, 四川成都 610031; 2. 宜宾职业技术学院智能制造学院, 四川宜宾 644003; 3. 中国航空工业集团公司洛阳电光设备研究所, 河南洛阳 471000; 4. 西南交通大学机械工程学院, 四川成都 610031)

摘要: 针对大型装备制造车间人机双资源调度中工人疲劳和学习效应集成优化难的问题, 基于指数型疲劳表征和 DeJong 学习曲线, 构建混合整数规划模型。根据双资源调度问题的编码特征, 结合排序规则生成初始调度方案, 结合移除和插入操作设计自适应大邻域搜索算法, 以求解人机双资源分配、工件排序、动态调度等子问题。同时, 围绕紧急插单、机器故障、工人离岗等扰动设计重调度策略, 基于原有算法实现动态调度, 并进行算法测试。结果表明, 与求解器、规则、遗传算法等相比, 所提算法找到了完工时间更短的调度方案, 且在紧急插单、机器故障、工人离岗等动态调度场景下, 也能够给出更好的结果。所研究的装备制造车间调度问题同时考虑了工人疲劳与学习效应, 对应算法可为大型装备制造排产系统开发提供参考。

关键词: 计算机辅助制造; 作业车间调度; 工人疲劳; 学习效应; 自适应大邻域搜索; 动态调度

中图分类号: TP399 **文献标识码:** A **DOI:** 10.7535/hbkd.2023yx02006

Equipment manufacturing job shop dynamic scheduling with worker fatigue and learning effect

GUO Chao^{1,2}, HAO Donghui³, GUO Peng^{1,4}

(1. Technology and Equipment of Rail Transit Operation and Maintenance Key Laboratory of Sichuan Province, Chengdu, Sichuan 610031, China; 2. School of Intelligent Manufacturing, Yibin Vocational and Technical College, Yibin, Sichuan 644003, China; 3. Luoyang Institute of Electro-Optical Equipment of AVIC, Luoyang, Henan 471000, China; 4. School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, Sichuan 610031, China)

收稿日期: 2023-02-18; 修回日期: 2023-03-18; 责任编辑: 王淑霞

基金项目: 国家重点研发计划项目(2020YFB1712200); 宜宾职业技术学院科研平台建设计划资助项目(YBZY21KYPT-03); 轨道交通运维技术与装备四川省重点实验室开放课题(2020YW004)

第一作者简介: 郭超(1984—), 男, 四川宜宾人, 讲师, 主要从事智能制造方面的研究。

通信作者: 郭鹏副教授。E-mail: pengguo318@swjtu.edu.cn

郭超, 郝东辉, 郭鹏. 考虑工人疲劳和学习效应的装备制造车间动态调度[J]. 河北科技大学学报, 2023, 44(2): 152-164.

GUO Chao, HAO Donghui, GUO Peng. Equipment manufacturing job shop dynamic scheduling with worker fatigue and learning effect[J]. Journal of Hebei University of Science and Technology, 2023, 44(2): 152-164.

Abstract: Aiming at the problem of integrating optimization of worker fatigue and learning effects faced by human and machine dual-resource scheduling in large equipment manufacturing workshops, a mixed-integer programming model was constructed based on exponential fatigue characterization and DeJong learning curves. In accordance with the coding characteristics of the dual-resource scheduling problem, an initial scheduling scheme was generated based on sorting rules, and an adaptive large neighborhood search algorithm was designed by combining removal and insertion operations to solve sub-problems, such as the allocation of human and machine resources, job sequencing and dynamic scheduling. A rescheduling strategy was also designed to address disturbances such as urgent job insertions, machine failures and worker absences. Based on the original algorithm, dynamic scheduling was implemented and tested. The results show that compared with solvers, rules, genetic algorithms and other methods, the proposed algorithm can find scheduling solutions with shorter completion times, and it also performs better in dynamic scheduling scenarios such as urgent orders insert, machine breakdowns and worker absences. The proposed equipment manufacturing workshop scheduling problem takes into account worker fatigue and learning effects, and the corresponding algorithm can provide reference for the development of large-scale equipment manufacturing production scheduling systems.

Keywords: computer aided manufacturing; job shop scheduling; worker fatigue; learning effect; adaptive large neighborhood search; dynamic scheduling

随着“中国制造 2025”战略的实施与推进,人机协同成为智能制造的主流发展趋势。一方面,在大型装备制造过程中,复杂的焊接或装配工艺导致难以依靠机器人自动化实现,高素质工人成为此类企业关键的生产资源^[1]。但过度的疲劳会对工人的身体健康产生不利的影响,甚至会造成严重的安全事故和经济损失^[2]。在车间调度问题中考虑工人的疲劳因素是十分必要的。另一方面,在生产过程中往往存在着工件动态到达、插单、机器故障等扰动事件,系统则需要根据扰动事件类型对初始调度方案作出调整以满足加工需求,通过合理的工作分配和休息策略,可以在保障工人安全生产的同时有效缩短生产周期、提高资源利用率。此外,学习效应同样是影响人机双资源柔性作业车间调度问题的重要因素。

JANSEN 等^[3]探究了不同工作制模式与员工疲劳患病率之间的关系,发现疲劳因素是工人辞职或转行的重要原因。LI 等^[4]针对工人疲劳易引起员工绩效下降的问题,提出基于休息的数学规划模型,有效提高了员工的满意度。JAMSHIDI 等^[5]强调人在调度系统中的维护作用,并针对工人疲劳提出基于疲劳恢复的数学模型,该模型中疲劳是当前工作时间和最大工作时间的比值。BERTI 等^[6]将人体的累积疲劳整合到调度算法中,让工人在疲劳达到一定强度时进行休息。MOUAYNI 等^[7]考虑了调度作业中的人为因素,基于贪婪随机自适应搜索过程,提出考虑工人疲劳和可靠性的整数规划作业调度模型。FU 等^[8]考虑了源自生物数学模型的生物数学疲劳约束,开发了嵌入禁忌搜索启发式调度算法以解决车辆路径和调度的组合问题。ZHANG 等^[9]针对人机协同装配任务调度中人的疲劳问题,提出了基于疲劳积累恢复特性的任务调度模型和改进的化学反应优化算法。尽管目前企业和学者对工人疲劳的处理措施多样,但仍然需要开发有效的调度算法来解决此类问题。

在学习效应方面,WRIGHT^[10]提出了对数线性模型以表示工人的学习效应,该模型中生产时间随着生产次数的增加而减小。在后续的数十年中,许多关于学习效应的研究都是基于此模型的改进,BISKUP^[11]总结了现有文献中关于学习模型的多种变体。JANIAK 等^[12]考虑了基于经验的学习模式。WANG 等^[13]提出工件实际加工时间与序列位置和位置权重相关的学习模型。此外,考虑医护人员疲劳效应和截断学习效应的医护调度^[14],加工时间可变情况下具有指数学习效应和凸资源分配排序^[15],考虑学习效应的双资源约束柔性车间调度^[16]等问题均有涉及。但在实际生产中,作业的实际处理时间不会随着操作次数的无限增加而趋于零,现有研究大多数忽略了学习效应,或未注意到学习效应存在着极限的问题,这将对生产制造系统的长时间调度产生较大的影响。

为此,本研究面向装备制造研究人机双资源约束柔性作业车间调度问题(dual resource constrained flexible job-shop scheduling problem, DRCFJSP),综合考虑工人的疲劳效应和学习效应,以最小化完工时间为优化目标,建立混合整数规划模型;针对调度过程中动态扰动事件,提出动态调度策略,并设计多种动态场景验证模型的响应能力;提出自适应大规模邻域搜索(adaptive large neighborhood search, ALNS)算法用于静态调度和动态调度的方案获取。

1 问题描述与模型建立

1.1 疲劳效应与学习效应

在生产调度过程中考虑工人疲劳因素,有利于保护工人健康和提高生产效率。文献[1]提出的指数型疲劳模型能较好反映出工人在实际工作过程中的疲劳变化情况,计算方式如下。

$$f_{t+1} = x_t \times (1 - (1 - f_t) \times \exp(-\lambda)) + (1 - x_t) \times f_t \times \exp(-\mu). \quad (1)$$

式中: f_t 和 f_{t+1} 分别为工人在 t 和 $t+1$ 时刻末的疲劳值; λ 和 μ 是工人疲劳累积率和疲劳恢复率; x_t 取值为0或1,1表示工人处在工作状态,0表示工人处在休息状态。而在生产加工过程中,适当休息有利于整个生产效率提升和避免工人累计疲劳损伤。而工人的休息时间最好由工人当前疲劳值和下一个要加工工件的加工时间动态确定。在此假设工人疲劳的休息标准为 f_e ,工人当前疲劳值为 f_0 ($f_0 < f_e$),工人加工下一工件的加工时长为 t_w 。首先,计算工人的可继续加工时间 t_c :

$$t_c = \frac{1}{\lambda} \times \ln \frac{1 - f_0}{1 - f_e}, \quad (2)$$

若 $t_c \geq t_w$,工人可继续工作;否则工人需要休息,休息时长 τ 为

$$f_e = 1 - (1 - f_1) \times \exp(-\mu \times t_w), \quad (3)$$

式中: $f_1 = f_0 \times \exp(-\mu \times \tau)$; $\tau = (\ln f_0 - \ln f_1) / \mu$ 。

在考虑学习效应的加工环境中,工件的实际加工时间并不会无限减少至0。在此采用DeJong学习模型^[17]对其进行描述如下:

$$T_x = T_0 \times (Z + (1 - Z) \times x^{-\varphi}), \quad (4)$$

式中: T_0 为某个工件的理论加工时长; T_x 为第 x 次加工此工件时的实际加工时长; Z 是取值在0到1之间的不可压缩系数,表征加工此工件的最小必需时长为 $Z \times T_0$; φ 是与工人的个人素质有关的学习效应系数,即学习能力的大小。

1.2 问题描述

假设某人机协作生产车间中有一定数量的工人和机器,分别用集合 $W = \{W_1, W_2, \dots, W_c\}$ 和集合 $M = \{M_1, M_2, \dots, M_d\}$ 表示,其中 c 和 d 分别表示工人总数和机器总数。同类型的机器具有相同的加工效率,不同类型的机器数量和效率不尽相同。由于职责和分工不同,工人所能操作的机器类型也不同。

现在计划加工一批共计 n 个工件,用集合表示为 $J = \{J_1, J_2, \dots, J_n\}$,由于机器类别和工人效率的不同,每个工件的实际加工时长取决于分配的机器资源和工人资源。需要满足下述约束条件。

1)工人的状态分为工作和休息2种;若为工作状态,工人只允许在一台机器上加工1个工件;若为休息状态,则不占用机器和工件。

2)机器的状态分为工作和不工作,若为工作状态,每台机器在同一时刻只允许加工1个工件,只允许被1个工人操作;若为不工作状态,则不占用工人和工件。

3)每个工件有2道工序:首先,需要工人在机器上辅助安装加工模具和上料;然后,再由机器自动加工,即工人处理部分和机器加工部分,两者之间不允许存在时间间隙。

4)每个工件都需要被加工1次。

5)某个工件一旦开始加工,负责处理的工人不能离开,直至工件的工人处理部分完成,负责加工的机器不能中断,直至当前工件的机器加工部分结束,除非发生动态事件。

6)所有工人和所有机器在整个加工过程中都是可用的,除非发生动态事件。

7)每台机器的位置是固定的,忽略工件以及工人在不同机器之间的转移时间。

8)所有工人和所有机器的初始开始时刻为零时刻。

9)在整个生产过程中,工人的疲劳值不允许超过指定上限。

10)工件的实际加工时间由工人、机器和加工顺序共同决定。

调度的目标是,在满足上述所有约束的条件下,为这批工件合理地分配机器和工人,确定加工顺序,以及每个工件加工的开始时间和结束时间,最终使得所有工件加工完毕时的完工时间最小化。

1.3 模型建立

基于上述问题的描述与假设,构建如下的混合整数规划模型,如表1所示。

表 1 模型的符号与定义
Tab.1 Symbols and definitions of models

符号	定 义
C_{\max}	最大完工时间
S	加工顺序集合 $S = \{1, 2, \dots, s, \dots, a\}$, 此处 a 表示工序总数
J	工件集合 $J = \{1, 2, \dots, j, \dots, n\}$, 此处 n 表示工件总数
W	工人集合 $W = \{1, 2, \dots, w, \dots, c\}$, 此处 c 表示工人总数
M	机器集合 $M = \{1, 2, \dots, m, \dots, d\}$, 此处 d 表示机器总数
b_{wm}	工人 w 是否可以操作机器 m , 是为 1, 否则为 0
t_{jm1}	工件 j 在机器 m 上加工时需要工人处理的理论时间
t_{jm2}	工件 j 在机器 m 上加工时需要机器加工的时间
λ_w	工人 w 的疲劳累计率
μ_w	工人 w 的疲劳恢复率
f_e	工人的疲劳上限
φ_w	工人 w 的学习效应系数
Z	学习效应的不可压缩系数
d_{swm}	在顺序 s 上由工人 w 和机器 m 加工时的工件开始时间
g_{sw}	在顺序 s 上工人 w 的结束时间
h_{sm}	在顺序 s 上机器 m 的结束时间
p_{sw}	在顺序 s 上工人 w 的工作时间
q_{sw}	在顺序 s 上工人 w 的休息时间
u_{sw}	在顺序 s 上工人 w 工作后的疲劳值
v_{sw}	在顺序 s 上工人 w 休息后的疲劳值
Y_{sw}	在顺序 s 上工人 w 的已工作次数, 包含本次
y_{sw}	在顺序 s 上工人 w 的实际学习率
x_{sjwm}	在顺序 s 上是否由工人 w 和机器 m 加工工件 j , 是为 1, 否则为 0

目标函数为

$$\min C_{\max} \quad (5)$$

约束为

$$\sum_{j \in J} \sum_{w \in W} \sum_{m \in M} x_{sjwm} = 1, \quad \forall s \in S, \quad (6)$$

$$\sum_{s \in S} \sum_{w \in W} \sum_{m \in M} x_{sjwm} = 1, \quad \forall j \in J, \quad (7)$$

$$\sum_{j \in J} \sum_{w \in W} \sum_{m \in M} x_{sjwm} \times b_{wm} = 1, \quad \forall s \in S, \quad (8)$$

$$d_{swm} \geq h'_{s'm}, \quad \forall s, s' \in S, s' < s, w \in W, m \in M, \quad (9)$$

$$d_{swm} \geq g'_{s'w}, \quad \forall s, s' \in S, s' < s, w \in W, m \in M, \quad (10)$$

$$Y_{sw} = \sum_{s' \in S, s' \leq s} \sum_{j \in J} \sum_{m \in M} x_{sjwm}, \quad \forall s \in S, w \in W, \quad (11)$$

$$y_{sw} = Z + (1 - Z) \times (Y_{sw})^{-\varphi_w}, \quad \forall s \in S, w \in W, \quad (12)$$

$$g_{sw} \geq \sum_{j \in J} \sum_{m \in M} (d_{swm} + t_{jm1} \times y_{sw}) \times x_{sjwm}, \quad \forall s \in S, w \in W, \quad (13)$$

$$h_{sm} \geq \sum_{j \in J} \sum_{w \in W} (d_{swm} + t_{jm1} \times y_{sw} + t_{jm2}) \times x_{sjwm}, \quad \forall s \in S, m \in M, \quad (14)$$

$$p_{sw} \geq \sum_{j \in J} \sum_{m \in M} t_{jm1} \times x_{sjwm}, \quad \forall s \in S, w \in W, \quad (15)$$

$$q_{sw} \geq \sum_{j \in J} \sum_{m \in M} (d_{swm} - g_{(s-1)wm}) \times x_{sjwm}, \quad \forall s \in S, w \in W, \quad (16)$$

$$v_{sw} = u_{(s-1)w} \times \exp(-\mu_w \times q_{sw}), \quad \forall s \in S, w \in W, \quad (17)$$

$$u_{sw} = 1 - (1 - v_{sw}) \times \exp(-\lambda_w \times p_{sw}), \quad \forall s \in S, w \in W, \quad (18)$$

$$u_{sw} \leq f_e, \quad \forall s \in S, w \in W. \quad (19)$$

式(5)为最小化完工时间的目标函数;约束(6)确保每台机器在同一时刻只能分配一个工人和一个工件;约束(7)确保每个工件仅可被加工一次;约束(8)确保工人只能操作自己可操作的机器;约束(9)和约束(10)要求工件的开始加工时间不能早于机器和工人的可开始时间;约束(11)记录了工人的实时工作次数;约束(12)根据工人的工作次数计算了工人的实时学习率;约束(13)和约束(14)分别计算了考虑学习效应的工人结束时间和机器结束时间;约束(15)和约束(16)计算了工人的工作时间和休息时间;约束(17)和约束(18)基

于疲劳公式(1)分别计算出了工人在休息后和工作后的疲劳值;约束(19)要求工人的疲劳值不超过规定上限。

2 改进自适应大邻域搜索算法

标准自适应大邻域搜索算法在解决车辆路径类问题^[18-20](vehicle routing problem, VRP)时表现出了优异的搜索效率和寻优性能。但当用来求解 DRCFJSP 时,破坏算子和修复算子的权重和得分只有在获得更优解后才得到更新,同时由于随机因素的存在,算法在每次迭代中并不能准确地选择到当前最优的修复算子,从而导致算法在搜索前期不能取得较好的效果,且容易在获得当前邻域局部最优解之前过早进入下一邻域,即早熟问题。针对这一问题,本文提出改进的 ALNS 算法,改进算法使用多组破坏算子和修复算子,每次迭代根据历史表现对破坏算子进行自适应选择,然后挑选修复算子中最优的一个进行修复操作,以获得新解。同时为了避免陷入局部最优,使用改进接受准则在一定概率下接受劣解。

2.1 编码与解码

双资源约束的调度问题可细分为工件排序、设备分配、工人指派 3 个子问题,相对应地,其可行解可用一定规则的三级编码来表达,3 个编码的长度相同且一一对应:一级是机器分配编码,其数字表示所选择的机器序号;二级是工人指派编码,其数字表示所选择的工人编号;三级是工件排序编码,其数字表示所选择的工件编号。图 1 展示了 2 个工人操作 4 台设备加工 10 个工件时的示例编码,以工件码第 4 位置的数字 6 为例,其表示工件 6 的加工顺序为第 4,且分配的资源是机器 2 和工人 1。

解码是将编码转化为目标函数的过程。首先,根据工序码中工件的顺序,依次为当前工件分配对应的机器和工人;然后,由所选机器和工人的最早可开始时间,确定当前工件的开始加工时间和实际加工时长,直至最后一个工件;最后,获得最大完工时间,即目标函数。

机器码	3	2	4	2	1	1	4	3	4	1
人工码	1	2	1	1	2	2	1	1	2	2
工件码	8	2	9	6	1	7	4	3	10	5

图 1 可行解编码图

Fig. 1 Feasible solution coding diagram

2.2 初始解生成

ALNS 算法需要一个初始解,其质量在一定程度上决定算法后续的优化效率。采用启发式规则方法来生成初始解,在学习效应的影响下,随着工人操作熟练度的增加,理论时长越长的工件排序越靠后,其所减免的加工时间就越多,这将有利于目标函数。初始解生成规则如图 2 所示,首先,选择最早的可用机器;其次,再选择当前机器所在单元的空闲工人中疲劳值最小的工人;最后,分配工人操作阶段时间最小的工件,若工件的工人操作时间相同,则选择机器处理时间最小的工件进行加工。

2.3 操作算子

在初始解生成后,开始迭代搜索。在每次迭代中,破坏算子会先对当前解移除一定数量的工件节点,得到破坏解和移除列表,然后修复算子再将移除列表中的节点按照一定规则插入破坏解中,最终得到新的可行解,如图 3 所示。改进的自适应大规模邻域搜索算法设计了 6 个破坏算子和 6 个修复算子。

2.3.1 破坏算子

1) 随机破坏算子:随机选择原序列中的 k 个工件进行移除,以此防止局部最优,在修复算子时能够对选定的 k 个工件进行重排序操作;

2) 贪婪破坏算子 1:依次计算当前工件移除后对原序列成本的影响,取对完工时间影响最大的前 k 个工件作为最终要移除的工件;

3) 贪婪破坏算子 2:依次计算当前工件移除后对原序列成本的影响,取对完工时间影响最大的工件作为当前要移除的工件,更新当前序列并重复此步骤 k 次;

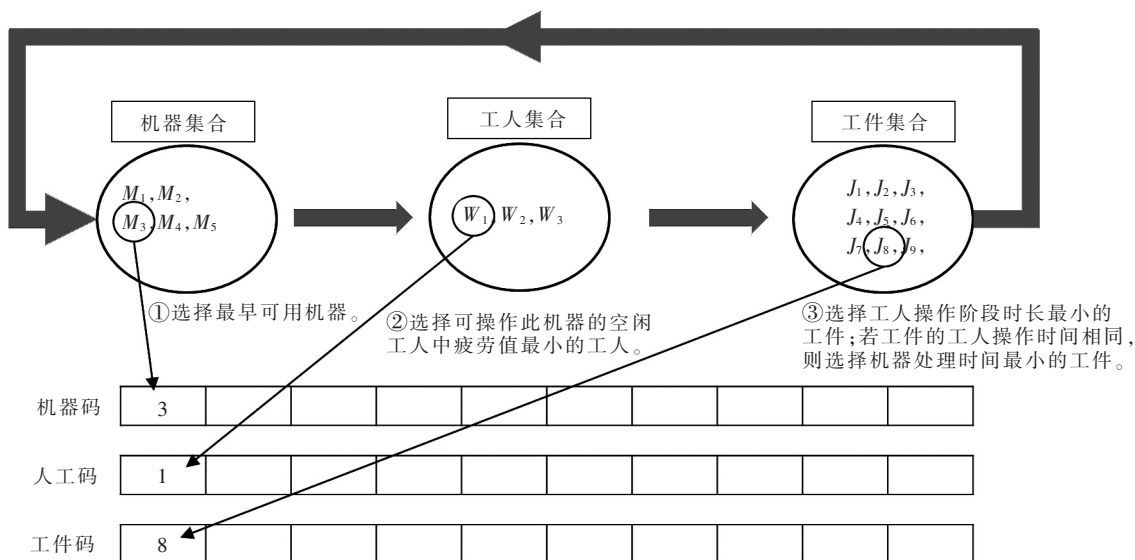


图 2 规则生成初始解

Fig. 2 Rule of generating initial solution

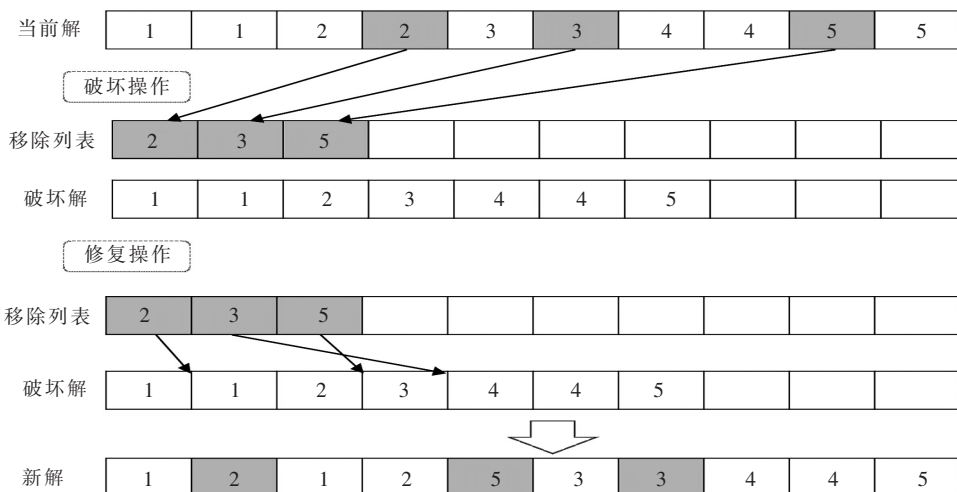


图 3 破坏和修复过程

Fig. 3 Destruction and repair process

4)相似破坏算子:随机移除一个工件,然后计算移除工件对剩余工件的相似度,再移除相似度较高的 $k-1$ 个工件,共计 k 个;

5)最差破坏算子 1:与贪婪破坏算子 1 相似,不同的是,移除的是对完工时间影响最小的工件;

6)最差破坏算子 2:与贪婪破坏算子 2 相似,不同的是,移除的是对完工时间影响最小的工件。

2.3.2 修复算子

1)随机修复算子:随机选择 k 个位置,将移除的工件依次插入;

2)最优修复算子:从移除列表的第一个工件开始,每次选择完工时间最小的位置进行插入,直至所有工件插入完毕;

3)随机最优修复算子:与最优修复算子相似,但移除工件列表要先随机打乱顺序;

4)次优修复算子:与最优修复算子类似,但插入的位置是完工时间第 2 小的位置;

5)后悔修复算子:首先,计算每个移除工件的最优插入和次优插入的成本,后悔值即两者的完工时间差,根据后悔值由大到小对移除工件列表排序,然后按照最优修复算子进行相同的操作;

6)逆序最优修复算子:与最优修复算子相似,但移除工件列表要进行逆序操作。

2.4 接受准则

为了避免陷入局部最优,本文设计了改进的接受准则:

$$r \leq e^{-\frac{(f_N - f_C)}{f_C}}, \quad (20)$$

式中: r 为0至1之间的随机数; f_N 和 f_C 分别是新解和当前解的目标函数值,即完工时间。该接受准则与模拟退火接受准则在形式上相似,但消除了温度参数,避免了复杂的参数调试过程。以新解与当前解的相对偏差作为衡量标准,可有效避免算法陷入局部最优,且相对偏差的值越小,接受新解的概率越大。

2.5 改进 ALNS 算法流程

本文针对标准 ALNS 算法求解 DRCFJSP 问题时早熟和不稳定的问题,提出改进的 ALNS 算法。改进算法使用启发式规则生成初始解,使用得分机制动态选择破坏算子,并挑选最有利于当前解的修复算子,使得在每次迭代中都能找到较好的破坏和修复的组合方式。若新解质量不如当前解,使用基于相对偏差的接受准则在一定概率下接受劣解继续寻优。算法整体步骤如表 2 所示。

表 2 改进的自适应大邻域搜索算法伪代码

Tab. 2 Pseudo code of improved ALNS algorithm

改进的自适应大邻域搜索算法伪代码	
输入:	工件信息、工人信息等相关参数
输出:	全局最优解 B
1:	使用启发式规则获得初始解 S
2:	令全局最优解 $B=S$ 和当前解 $C=S$
3:	初始化破坏算子的使用次数、得分、权重等参数
4:	while 当前迭代次数 \leq 终止迭代次数 do
5:	自适应选择破坏算子,更新破坏算子的使用次数
6:	对当前解 C 依次进行破坏操作
7:	挑选最优的修复算子修复当前解,并生成新解 N
8:	if $f_N < f_C$ do
9:	更新当前解 C 和破坏算子的分数
10:	if $f_N < f_B$ do
11:	更新全局最优解 B 和破坏算子的分数
12:	end if
13:	else do
14:	if 满足改进接受准则 do
15:	更新当前解 C 和破坏算子的分数
16:	end if
17:	end if
18:	更新算子权重和当前迭代次数
19:	end while
20:	输出全局最优解 B

3 静态调度

本文算法编程的计算机硬件配置为 Intel i7-12700H CPU 和 16 GB RAM,计算机软件环境为 Windows 11 的操作系统和 Python 3.9 语言。

3.1 算例生成

为了验证本文所提改进自适应大邻域搜索算法在求解考虑工人疲劳的双资源柔性作业车间调度问题时的合理性和有效性,本文结合文献^[1]中的算例并进行了扩展,包含小、中、大和超大 4 种规模共计 16 个算例,分别命名为 D1 至 D16。同时,设置 200 个不同的工件(编号分别为 J_1 到 J_{200})以供选择,其中工件的加工时长、工人的疲劳率、工人的疲劳上限、工人学习效应不可压缩系数与工人的学习效应系数等信息均通过随机的方法生成(具体的算例数据联系作者可获取)。算例信息如表 3 所示。

表 3 算例数据表

Tab. 3 Data table of calculation examples

规模	算例	工件总数	工件分配	机器类型	机器总数	机器分配	工人总数	工人分配
小规模	D1	10	$J_{11}-J_{20}$	2	6	[4,2]	3	[2,1]
	D2	10	$J_{31}-J_{40}$	2	6	[4,2]	3	[2,1]
	D3	15	$J_{16}-J_{30}$	2	6	[4,2]	3	[2,1]
	D4	15	$J_{46}-J_{60}$	2	6	[4,2]	3	[2,1]
中规模	D5	25	$J_{26}-J_{50}$	3	10	[4,4,2]	5	[2,2,1]
	D6	25	$J_{76}-J_{100}$	3	10	[4,4,2]	5	[2,2,1]
	D7	30	$J_{31}-J_{60}$	3	10	[4,4,2]	5	[2,2,1]
	D8	30	$J_{91}-J_{120}$	3	10	[4,4,2]	5	[2,2,1]
大规模	D9	40	$J_{41}-J_{80}$	4	14	[5,4,3,2]	8	[3,2,2,1]
	D10	40	$J_{121}-J_{160}$	4	14	[5,4,3,2]	8	[3,2,2,1]
	D11	50	$J_{51}-J_{100}$	4	14	[5,4,3,2]	8	[3,2,2,1]
	D12	50	$J_{151}-J_{200}$	4	14	[5,4,3,2]	8	[3,2,2,1]
超大规模	D13	70	$J_{71}-J_{140}$	5	20	[6,5,4,3,2]	11	[3,3,2,2,1]
	D14	70	$J_{131}-J_{200}$	5	20	[6,5,4,3,2]	11	[3,3,2,2,1]
	D15	90	$J_{41}-J_{130}$	5	20	[6,5,4,3,2]	11	[3,3,2,2,1]
	D16	90	$J_{111}-J_{200}$	5	20	[6,5,4,3,2]	11	[3,3,2,2,1]

3.2 参数调试

算法参数的设置会影响算法的收敛速度和求解质量,改进的 ALNS 算法主要涉及迭代次数和破坏算子移除的工件数量 k 2 个关键参数,接下来需要对其进行调试。从小、中、大、超大 4 个规模中各随机选择 1 个算例,测算了最大迭代次数为 300 时 k 值分别为 1 至 5 的表现,结果如图 4 所示。由图 4 可以看出,无论 k 取何值时,迭代次数在 100 次之后完工时间已趋于稳定,考虑到过多的迭代次数会有较高的时间成本,因此,选择迭代次数为 100。同时,实验运行时间如图 5 所示。

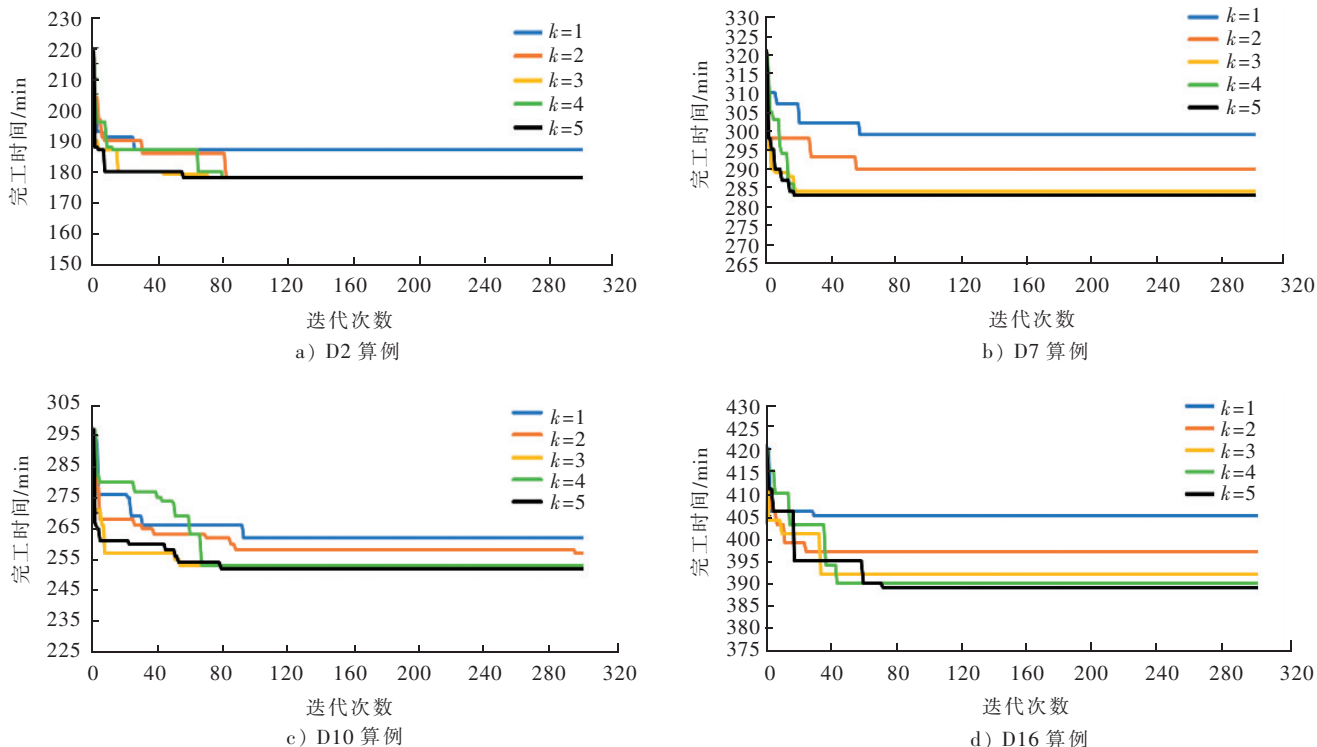


图 4 迭代收敛曲线

Fig. 4 Iterative convergence curve

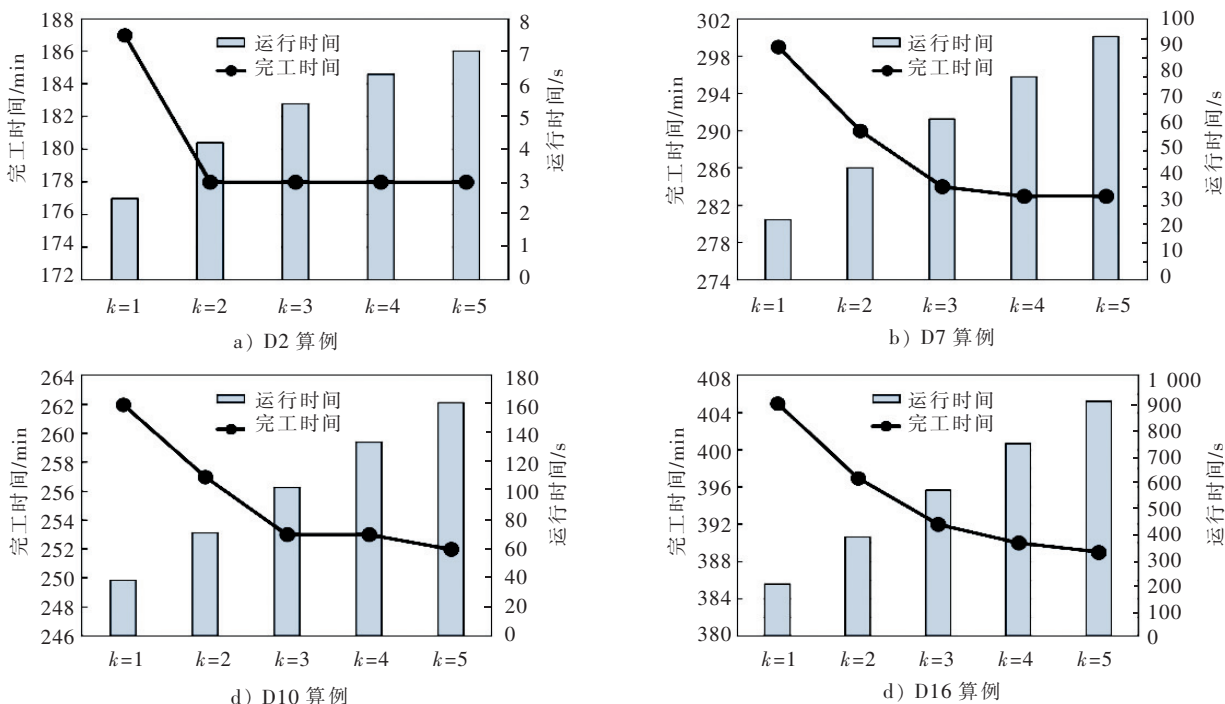


图 5 完工时间和运行时间对照图

Fig. 5 Comparison charts of makespan and runtime

从图 5 中的数据可以看出程序运行时间随着 k 值的增大呈线性增长;本文的优化目标是 minimized 完工时间,数据显示随着 k 值的增大目标函数值的提升空间逐渐缩小,为了避免过多的时间消耗,选择 k 值为 3,此时有较好的寻优效果和适中的运行时间。为此,后续计算中涉及的算法参数设置为:迭代次数为 100,破坏算子移除的工件数量 k 为 3。

3.3 结果与对比

由于算法存在随机因素,为了减少随机误差的影响,每个算例重复独立运行 10 次,并使用平均值、最优值、标准差、变异系数和平均运行时间 5 个指标来衡量算法的性能,其中变异系数是衡量实验样本偏离平均值的程度。实验结果如表 4 所示。从表 4 可以看出,计算结果的标准差和变异系数都在较小范围内,同一规模下的程序运行时间也都保持在相同水平,这体现了本文改进算法在处理复杂调度问题时的优化性能和稳定性能,能够较好处理不同规模的问题。

表 4 算例计算结果

Tab. 4 Calculation results of calculation example

算例	最优值	平均值	标准差	变异系数	平均耗时/s
D1	172	172.8	0.40	0.002	1.0
D2	162	162.8	0.40	0.002	1.0
D3	236	237.2	0.75	0.003	2.5
D4	203	205.4	1.20	0.006	2.5
D5	213	216.6	1.80	0.008	7.6
D6	210	215.7	2.76	0.013	7.6
D7	245	248.3	1.55	0.006	11.2
D8	243	248.5	2.33	0.009	10.5
D9	232	235.6	1.62	0.007	21.4
D10	233	239.2	2.18	0.009	21.6
D11	279	282.0	3.55	0.013	33.7
D12	284	287.8	2.14	0.007	34.0
D13	274	279.5	2.06	0.007	71.8
D14	286	290.2	2.14	0.007	72.5
D15	338	341.7	2.61	0.008	119.5
D16	344	349.4	3.17	0.009	120.6

为了深入分析问题,使用 Gurobi 求解器来求解数学规划模型。生成规则具有计算量低、速度快的优点,但求解质量往往不高,通常用作生成初始解;遗传算法是应用较为广泛的经典启发式算法;Jaya 算法求解柔性作业车间调度问题^[21],表现出较好的性能。因此,将 Jaya 算法作为求解 DRCFJSP 的对比算法之一;标准 ALNS 算法则用于比较所提算法改进部分的优势。本文使用以上 5 种不同的算法作为对比,以进一步验证改进算法的性能和效率。其中,设置 Gurobi 的最大求解时间为 5 h。表 5 为最终得到的实验结果,其中“—”表示 Gurobi 未在指定时间内求得可行解。

表 5 算法测算结果对比表

Tab. 5 Comparison of algorithm calculation results

算例	Gurobi 求解器		生成规则		遗传算法		Jaya 算法		标准 ALNS		改进 ALNS	
	结果/ min	耗时/ s	结果/ min	耗时/ s	结果/ min	耗时/ s	结果/ min	耗时/ s	结果/ min	耗时/ s	结果/ min	耗时/ s
D1	172	18 000	195	<1	173	2.3	173	9.2	173	0.2	172	1.0
D2	173	18 000	204	<1	163	2.2	163	9.1	164	0.2	162	1.0
D3	278	18 000	274	<1	236	3.4	247	18.5	239	0.7	236	2.5
D4	288	18 000	255	<1	213	3.3	210	18.1	210	0.7	203	2.5
D5	—	18 000	237	<1	224	5.5	225	49.2	220	1.9	213	7.6
D6	—	18 000	230	<1	221	5.8	224	50.6	222	2.0	210	7.6
D7	—	18 000	273	<1	254	6.7	261	71.1	249	2.8	245	11.2
D8	—	18 000	260	<1	260	6.7	266	65.7	257	2.5	243	10.5
D9	—	18 000	243	<1	249	9.7	248	130.4	239	5.0	232	21.4
D10	—	18 000	256	<1	254	9.8	243	131.4	241	5.8	233	21.6
D11	—	18 000	291	<1	298	12.0	291	198.2	291	8.3	279	33.7
D12	—	18 000	301	<1	304	12.4	304	199.1	297	7.8	284	34.0
D13	—	18 000	315	<1	296	18.6	298	415.7	276	18.7	274	71.8
D14	—	18 000	299	<1	305	18.4	309	424.4	296	17.5	286	72.5
D15	—	18 000	372	<1	371	23.9	375	695.1	354	28.8	338	119.5
D16	—	18 000	371	<1	380	23.9	381	716.3	360	28.8	344	120.6

表 5 结果显示,改进 ALNS 算法在多个算例的求解中都取得了最好的调度结果,表明改进算法具有一定的优越性;由于数学规划模型中加入了非线性的疲劳约束,在线性化处理的过程中需要引入多个中间变量,使得 Gurobi 算法的求解难度增大,在中大规模问题的求解中未获得可行解;相较于 Jaya 算法,改进算法具有较好的寻优速度;相较于生成规则、遗传算法和标准 ALNS 算法,改进算法具有较好的求解质量。数据共同表明本文改进算法的稳定性和有效性,具有求解质量高、适用规模大和求解稳定性强的优点。

4 动态调度

由于客户需求和生产环境的变化,实际生产过程存在着紧急插单、机器故障、工人离岗等多种不确定因素,极大扰乱生产线的效率以及交货期。因此,系统需要一定的重调度能力来应对此种不确定性的动态事件,以保证生产系统的稳定性。在此研究紧急插单、机器故障、工人离岗 3 个动态事件下的调度策略,以增强所提模型在应对扰动时的灵活性和鲁棒性,使之更贴合于实际。应对动态事件的重调度策略如图 6 所示。其中调度决策过程如下。

步骤 1:以动态发生时间为线,将工件分为已加工、正在加工、未加工 3 类;

步骤 2:遍历所有正在加工工件,更新当前工人、当前机器的可用时间为当前工件的完工时间;

步骤 3:判断当前工件是否在取消的订单中,是则执行步骤 4,否则跳过步骤 4;

步骤 4:停止此工件的加工,修改当前工人、当前机器的可用时间为当前时刻;

步骤 5:判断当前工件的加工工人是否为离岗工人,是则执行步骤 6 和步骤 7,否则跳过步骤 6 和步骤 7;

步骤 6:停止加工此工件,修改当前机器的可用时间为当前时刻,将此工件添加到未加工的工件集合中;

步骤 7:修改当前工人的可用时间为当前时刻+缺岗时长,并更新疲劳;

步骤 8:判断当前工件的加工机器是否为故障机器,是则执行步骤 9 和步骤 10,否则跳过步骤 9 和步骤 10;

步骤 9:停止此工件的加工,修改当前工人的可用时间为当前时刻,更新疲劳,并将此工件添加到未加工的工件集合中;

步骤 10:修改当前机器的可用时间为当前时刻加维修时长;

步骤 11:从未加工的工件中删除取消的工件,将插单的工件添加到未加工的工件集合;

步骤 12:获得新批次工件(未加工)、机器和工人的可开始时间,以及工人疲劳等。

本节获得初始调度方案和重调度方案的方法仍使用改进 ALNS 算法。

4.1 紧急插单场景

紧急插单除满足基础约束外,还需满足:1)插单时刻之前已开始加工但未完工的工件具有最高的加工优先权,其加工过程不允许中断,此工件工人工序的结束时间是该工人重调度的可开始时间,机器工序的结束时间是该机器重调度的可开始时间;2)插单时刻之前未开始加工的工件需要和插单工件一起进行重调度。

根据紧急插单事件的特点,设计了典型的插单场景:以 D8 算例为初始加工任务,在 $t=80$ 时刻增派工件集合为 $\{J_1, J_2, J_3, J_4, J_5, J_6, J_7, J_8, J_9, J_{10}\}$ 的订单。插单场景下的调度甘特图如图 7 所示。图 7 中 $0\sim 80$ min 之间是初始调度计划,在 80 min 发生了插单事件,以黑色竖线表示,此时新工件的插入并不会影响正在加工的工件;在 80 min 后,开始执行重调度方案,最终完工时间为 314 min。与其他算法的对比结果如表 6 所示,表 6 中的结果说明改进算法具有良好的寻优性能。

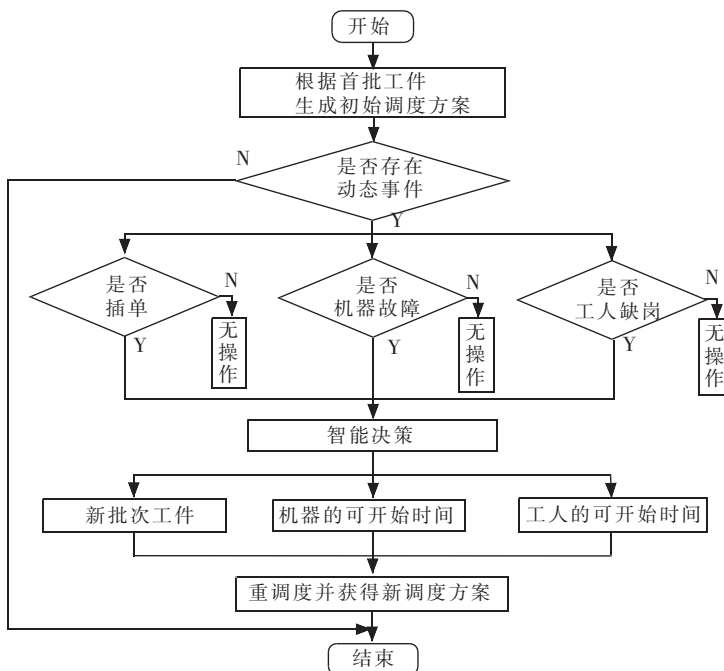


图 6 动态事件重调度策略

Fig. 6 Dynamic event rescheduling strategy

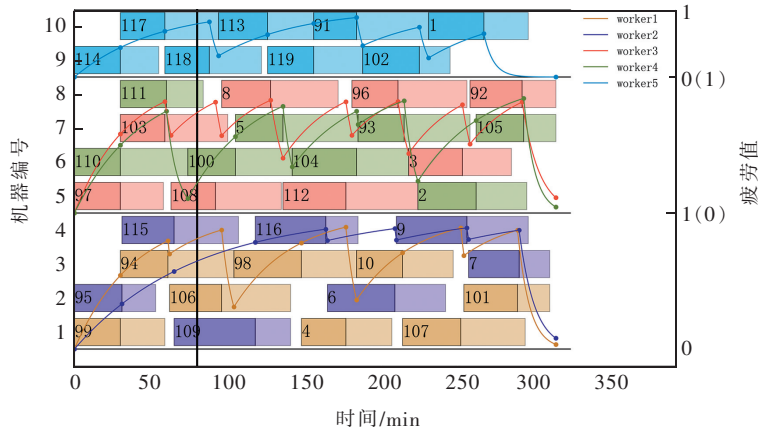


图 7 插单场景的调度甘特图

Fig. 7 Scheduling Gantt chart of job insertion scenarios

表 6 插单场景的调度结果

Tab. 6 Scheduling results of job insertion scenarios

策略	遗传算法	Jaya 算法	标准 ALNS	改进 ALNS
初始调度	260	266	257	243
重调度	327	328	324	314

4.2 机器故障场景

机器故障事件除了满足基础约束外,还需要满足:1)机器故障发生时,若此机器正在加工工件,需立即停止此工件的加工,由于此工件的加工被打断,需要重新开始加工;2)其他非故障机器的加工不受影响。

基于上述问题的描述与约束条件,构建的机器故障重调度场景为:以 D12 算例为初始加工任务,在 $t = 60$ 时刻机器 M_4 发生故障,且经过 50 min 维修后,在 $t = 110$ 时刻重新投入使用。机器故障场景下调度甘特图如图 8 所示。图 8 显示机器 M_4 在 60 min 发生了故障,此时正在加工的工件为 J_{165} ,由于加工中断,因此该工件需要重新加工,其重加工任务由工人 W_8 在机器 M_{12} 上执行;在经过 50 min 的维修后,机器 M_4 在 110 min 重新投入了使用,图中网状方块表示机器的不可使用时间。此外,与其他算法的对比结果如表 7 所示,数据表明了改进 ALNS 算法的优异性能。

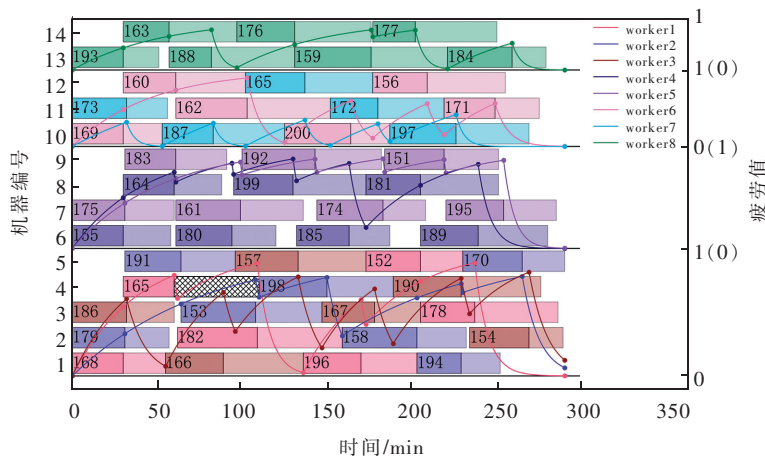


图 8 机器故障场景的调度甘特图

Fig. 8 Scheduling Gantt chart of machine fault scenario

表 7 机器故障场景的调度结果

Tab. 7 Scheduling results of machine fault scenarios

策略	遗传算法	Jaya 算法	标准 ALNS	改进 ALNS
初始调度	304	304	297	284
重调度	312	310	299	290

4.3 工人离岗场景

工人离岗事件除了必要的约束外,还需要满足:1)工人离岗事件发生时,此工人加工的工件按完成度可分为 4 种状态:若该工件还未开始加工,则需要安排重调度;若已完成工人处理部分和机器处理部分,则该工件不需要重新调度;若该工件处在工人加工部分且未完工,则立即停止此工件的加工,由于此工件的加工被打断,需要重新开始加工;若该工件已完成工人加工部分,处在机器自动处理阶段且未完工,由于机器自动处理阶段不需要工人操作,因此该工件不需要重新调度,待机器自动处理结束即可完成该工件的加工;2)其他非离岗工人的工作状态不受影响。

基于上述问题的描述与约束条件,构建了典型的工人离岗场景:以 D16 算例为初始加工任务,在 $t = 80 \text{ min}$ 工人 W_4 由于参加紧急会议而离岗,且经过 60 min 后,在 $t = 140 \text{ min}$ 重新进入工作岗位。工人离岗场景下调度甘特图如图 9 所示。

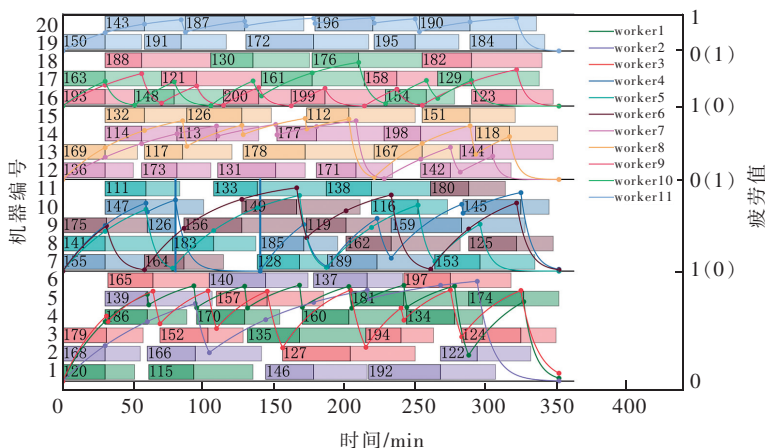


图 9 工人离岗场景调度甘特图

Fig. 9 Gantt chart of worker departure scenarios

图 9 中同颜色的 2 条竖线之间的区间表示的是工人 W_4 的不在岗时间。在离岗时刻,工件 J_{126} 的工人处理部分未完成,因此,需要重新安排调度,其重加工任务由工人 W_8 在机器 M_{15} 上执行;在 140 min 工人 W_4 回归岗位,开始加工的第一个工件是 J_{185} ,最终的完工时间为 352 min。与其他算法的对比结果如表 8 所示,表 8 的数据进一步说明本文改进算法可以有效地处理工人离岗动态事件,且具有一定的优越性。

表 8 工人离岗场景的调度结果

Tab. 8 Scheduling results of worker departure scenarios

	min			
策略	遗传算法	Jaya 算法	标准 ALNS	改进 ALNS
初始调度	380	381	360	344
重调度	387	391	368	352

5 结 语

本文基于双资源柔性作业车间调度问题,重点研究了调度过程中工人的疲劳和学习效应,以及动态调度问题,并建立了相应的混合整数规划模型。根据问题的特点,本文对标准的自适应大规模邻域搜索算法进行了合理改进,通过设置多组破坏算子和修复算子来寻求最早的完工时间,以提高人机协同调度过程中的资源利用率和生产效率,并保障工人的身心健康和安全。最终不同规模的多个算例实验结果验证了本文模型和改进算法的可行性和合理性,同时,紧急插单、机器故障、工人离岗复杂场景下的测试进一步验证了本文改进 ALNS 算法的正确性,有效解决了考虑工人疲劳和学习效应的双资源柔性作业车间的动态调度问题,为实际的人机协同车间生产系统提供了有力的理论基础。

未来将进一步优化算法的性能,提出更好的搜索方法提升算法的搜索效率。除此之外,将研究在更复杂约束条件下的 DRCFJSP,如工件随机到达、工件临时取消以及多目标求解等。

参考文献/References:

- [1] DU Hangming, QIAO Fei, WANG Junkai, et al. A hybrid metaheuristic algorithm with novel decoding methods for flexible flow shop scheduling considering human fatigue[C]//2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne; IEEE, 2021; 2328-2333.
- [2] BALAS J. How You Could Pay the Price for Exhausted Employees[EB/OL]. <https://www.constructionbusinessowner.com/safety/how-you-could-pay-price-exhausted-employees>, 2023-01-29.
- [3] JANSEN N W H, van AMELSVOORT L G P M, KRISTENSEN T S, et al. Work schedules and fatigue: a prospective cohort study[J]. Occupational and Environmental Medicine, 2003, 60(sup1): 47-53.
- [4] LI Kai, XU Shuling, FU Hong. Work-break scheduling with real-time fatigue effect and recovery[J]. International Journal of Production Research, 2020, 58(3): 689-702.
- [5] JAMSHIDI R, MAADI M. Maintenance and work-rest scheduling in human-machine system according to fatigue and reliability[J]. International Journal of Engineering(IJE), 2017, 30(1): 85-92.
- [6] BERTI N, ARTIGUES C, BATTÀIA O, et al. Heuristic approaches for scheduling manufacturing tasks while taking into account accumulated human fatigue[J]. IFAC-PapersOnLine, 2019, 52(13): 963-968.
- [7] MOUAYNI I E, DEMESURE G, HAOUZI H B E, et al. Jobs scheduling within Industry 4.0 with consideration of worker's fatigue and reliability using Greedy Randomized Adaptive Search Procedure[J]. IFAC-PapersOnLine, 2019, 52(19): 85-90.
- [8] FU Jiawei, MA Liang. Long-haul vehicle routing and scheduling with biomathematical fatigue constraints[J]. Transportation Science, 2022, 56(2): 404-435.
- [9] ZHANG Ming, LI Chunquan, SHANG Yuling, et al. Cycle time and human fatigue minimization for human-robot collaborative assembly cell[J]. IEEE Robotics and Automation Letters, 2022, 7(3): 6147-6154.
- [10] WRIGHT T P. Factors affecting the cost of airplanes[J]. Journal of the Aeronautical Sciences, 1936, 3(4): 122-128.
- [11] BISKUP D. A state-of-the-art review on scheduling with learning effects[J]. European Journal of Operational Research, 2008, 188(2): 315-329.
- [12] JANIAK A, RUDEK R. Experience-based approach to scheduling problems with the learning effect[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2009, 39(2): 344-357.
- [13] WANG Jibo, WANG Jianjun. Scheduling jobs with a general learning effect model[J]. Applied Mathematical Modelling, 2013, 37(4): 2364-2373.
- [14] 成舒凡, 叶春明. 嵌入疲劳效应和学习效应的应急手术调度研究[J]. 计算机工程与应用, 2022, 58(13): 295-302.
CHENG Shufan, YE Chunming. Study on emergency surgical scheduling with fatigue effect and learning effect[J]. Computer Engineering and Applications, 2022, 58(13): 295-302.
- [15] 杨舒涵, 王吉波. 带指数学习效应和凸资源分配的单机排序问题研究[J]. 重庆师范大学学报(自然科学版), 2022, 39(5): 24-28.
YANG Shuhan, WANG Jibo. Study on single machine scheduling problems with exponential learning effects and convex resource allocation [J]. Journal of Chongqing Normal University (Natural Science), 2022, 39(5): 24-28.
- [16] 侯天天, 张守京, 杜昊天. 考虑工人学习效应的双资源约束柔性车间调度[J]. 组合机床与自动化加工技术, 2022, 586(12): 164-168.
HOU Tiantian, ZHANG Shoujing, DU Haotian. Research on dual-resource constrained flexible job shop scheduling problem considering workers' learning effect[J]. Modular Machine Tool & Automatic Manufacturing Technique, 2022, 586(12): 164-168.
- [17] de JONG J R. The effects of increasing skill on cycle time and its consequences for time standards[J]. Ergonomics, 1957, 1(1): 51-60.
- [18] 伍国华, 毛妮, 徐彬杰, 等. 基于自适应大规模邻域搜索算法的多车辆与多无人机协同配送方法[J]. 控制与决策, 2023, 38(1): 201-210.
WU Guohua, MAO Ni, XU Binjie, et al. Research on the cooperative delivery of multiple vehicles and multiple drones based on adaptive large neighborhood search[J]. Control and Decision, 2023, 38(1): 201-210.
- [19] 徐倩, 熊俊, 杨珍花, 等. 基于自适应大邻域搜索算法的外卖配送车辆路径优化[J]. 工业工程与管理, 2021, 26(3): 115-122.
XU Qian, XIONG Jun, YANG Zhenhua, et al. Route optimization of takeout delivery vehicles based on adaptive large neighborhood search algorithm[J]. Industrial Engineering and Management, 2021, 26(3): 115-122.
- [20] MARA S T W, NORCAHYO R, JODIAWAN P, et al. A survey of adaptive large neighborhood search algorithms and applications[J]. Computers & Operations Research, 2022, 146. DOI:10.1016/j.cor.2022.105903.
- [21] 郭鹏, 赵文超, 雷坤. 基于改进 Jaya 算法的双资源约束柔性作业车间调度[J]. 吉林大学学报(工学版), 2023, 53(2): 480-487.
GUO Peng, ZHAO Wenchao, LEI Kun. Dual-resource constrained flexible job shop optimal scheduling based on an improved Jaya algorithm[J]. Journal of Jilin University (Engineering and Technology Edition), 2023, 53(2): 480-487.