

文章编号:1671-6833(2025)01-0017-09

组合缓冲约束下的多目标混合流水线节能调度

轩 华, 耿祝新, 李 冰

(郑州大学 管理学院, 河南 郑州 450001)

摘 要:为解决生产阶段间带有限缓冲和阻塞两种中间缓冲约束的混合流水线节能调度问题,考虑不相关并行机和多时间约束建立数学模型,结合问题特征提出一种改进多目标模因算法以同时最小化最大完工时间和机器总能耗。采用基于不相关机器分配的矩阵编码方案,利用基于 Tent 混沌映射的混合初始化策略生成初始元胞数组,全局优化算子应用基于参数的自适应遗传策略改进的非支配排序遗传算法,局部增强搜索算子应用一种融合自适应选择邻域搜索和多目标模拟退火的搜索策略以提高算法搜索能力。通过 24 种不同规模问题的算例实验,验证了所提算法求解该问题的有效性和优越性。实验结果表明:改进多目标模因算法在平均运行时间 241.26 s 内得到的平均 IGD 值为 47.89,平均 SP 值为 857.25,均低于其他 3 种对比算法。改进多目标模因算法所求解集具有较好的收敛性、多样性和分布性。

关键词:混合流水线;改进多目标模因算法;组合缓冲约束;不相关并行机;多目标优化;节能调度

中图分类号:TB49

文献标志码:A

doi:10.13705/j.issn.1671-6833.2024.04.009

混合流水线调度问题(hybrid flowline scheduling problem, HFSP)是一种典型的 NP-hard 问题。在实际工业中,由于存储设备、物理空间和特性等限制,某些特定生产阶段间的缓冲区往往是有限的,甚至不存在缓冲区。例如在苹果酒的制作过程中,过滤阶段需要去除果汁中的果肉,由于过滤机器容量有限,果汁会停留在发酵阶段等待过滤完成,而最后过滤和装瓶阶段之间则没有限制^[1]。由此可以提出一种部分生产阶段间存在无限缓冲区而其他阶段间没有缓冲区的混合流水线调度问题。

根据阶段间缓冲类型的不同,有关 HFSP 的研究可分为 4 种:带无限缓冲的 HFSP、阻塞 HFSP、带有限缓冲的 HFSP 和零等待 HFSP。针对带无限缓冲的 HFSP,王君妍等^[2]考虑工件动态到达、运输时间和机器调整时间,提出改进自适应遗传算法以最小化总加权完成时间。针对阻塞 HFSP, Qin 等^[3]考虑族顺序相关调整时间,提出迭代贪婪算法以解决集调度的 makespan 问题;Missaoui 等^[4]考虑交货时间窗,提出迭代贪婪算法以最小化总加权延迟和提前惩罚。针对带有限缓冲的 HFSP,胡蓉等^[5]提出

一种混合探路者算法以同时最小化 makespan 和总能量消耗;Lin 等^[6]研究了带工件序列和运输时间的可重入 HFSP,提出了混合和声搜索和遗传算法的启发式算法以同时最小化 makespan 和平均流程时间。针对零等待 HFSP,裴小兵等^[7]提出一种新型混合改进遗传算法以最小化 makespan;Azaiez 等^[8]研究了由医院手术室调度引起的带阶段灵活性的两阶段问题,提出 4 种下界和 4 个启发式以最小化 makespan。

针对组合缓冲约束的调度问题,在流水线环境下, Wang 等^[9]考虑无限缓冲和零等待约束假设每台机器上工件加工顺序相同,提出修正迭代贪婪算法;Riahi 等^[1]研究了无限缓冲和不同阻塞类型组合的调度问题,提出分散搜索算法以最小化 makespan。在混合流水线环境下,钟稜充等^[10]将集装箱码头调度视作三阶段混合零等待 HFSP,提出改进离散布谷鸟算法以最小化 makespan 和总作业成本。为解决以最小化 makespan 为目标的 HFSP,轩华等^[11]考虑不相关并行机环境下组合无限缓冲和零等待约束,提出一种混合离散人工蜂群算法;Zhuang

收稿日期:2024-02-07;修订日期:2024-04-15

基金项目:河南省科技研发计划联合基金资助项目(242103810046);河南省科技攻关计划项目(232102321093, 232102321026);河南省哲学社会科学规划项目(2023BJJ085)

作者简介:轩华(1979—),女,河南睢县人,郑州大学教授,博士,主要从事生产计划与调度、物流优化与控制等研究, E-mail: hxuan@zzu.edu.cn。

引用本文:轩华,耿祝新,李冰.组合缓冲约束下的多目标混合流水线节能调度[J].郑州大学学报(工学版),2025,46(1):17-25.(XUAN H, GENG Z X, LI B. Multi-objective hybrid flowline energy-saving scheduling with combined buffer constraints[J]. Journal of Zhengzhou University (Engineering Science), 2025, 46(1): 17-25.)

等^[12]将自动化集装箱码头的智能装卸设备的调度问题视作带双向流和有限缓冲的阻塞 HFSP,提出一种自适应大邻域搜索算法。

综上所述,现有研究已针对阶段间具有单一缓冲约束的 HFSP 取得了大量的成果,而阶段间组合不同中间缓冲约束的复杂车间环境下的 HFSP,尤其是多目标问题的研究则相对较少。探索节能减排机制以减少机器负载,平衡生产效率和能源消耗,对于 HFSP 的研究具有重要意义。

本文以最小化最大完工时间和机器总能耗为优化目标,研究不相关并行机环境下的生产阶段间既有无限缓冲情况又有阻塞情况的 HFSP (multi-objective hybrid flowline energy-saving scheduling problem with combined buffer constraints, MOHFESP-CBC),并提出一种改进多目标模因算法(improved multi-objective memetic algorithm, IMOMA)以近似求解该问题。

1 问题描述与模型

1.1 MOHFESP-CBC 描述

所研究 MOHFESP-CBC 描述如下。生产线中有 N 个工件需依次经过 S 个生产阶段,每个阶段 i 包含 U_i ($k=1,2,\dots,U_i$) 台不相关并行机。每个工件 j 有不同的动态到达时间记为 RT_j ($RT_j \geq 0$),即当工件到达生产系统后才能开始加工。工件 j 在生产阶段 i 的不同机器 k 上的加工时间取决于该工件以及用于加工该工件的机器。当工件完成前一阶段的加工后需要经运输后到达下一阶段才能开始加工。

上述 S 个生产阶段中,部分阶段间不存在缓冲区,即具有阻塞约束。将有阻塞约束的相邻生产阶段的前一阶段 i 记入阻塞阶段集,记为 W ,将集合 W 中所包含阶段数记为 s ,则无限缓冲阶段数为 $S-s$ 。

在实际的加工过程中,机器能耗分为加工能耗和待机能耗。为减少机器的待机时间,所有机器开始加工首道工序时开机,末道工序完工时关机,阻塞期间由于机器未进行加工,算作待机状态。调度的目标是确定工件在各阶段的加工序列及机器分配以同时最小化最大完工时间和机器总能耗。

1.2 模型构建

定义 U_i 为阶段 i 的不相关并行机数; P_{ijk} 为工件 j 在阶段 i 的机器 k 上的加工时间; TT_i 为阶段 i 到下一阶段的运输时间; E 为机器消耗总能量; E_{ik}^B 为阶段 i 的机器 k 加工所消耗的能量; E_{ik}^W 为阶段 i 的机器 k 待机所消耗的能量; V_{ik}^B 为阶段 i 的机器 k 的加工功率; V_{ik}^W 为阶段 i 的机器 k 的待机功率; T 为

总计划时间,时刻 $t \in \{1,2,\dots,T\}$; r 为阶段 i 的机器 k 的加工工序序号, $r \in \{1,2,\dots,m_{ik}\}$; H 为一个很大的正数。

定义决策变量 CT_{ij} 为工件 j 在阶段 i 的完工时间; BT_{ij} 为工件 j 在阶段 i 的开工时间; C_{ik}^r 为阶段 i 的机器 k 的第 r 道工序的完工时间; B_{ik}^r 为阶段 i 的机器 k 的第 r 道工序的开始时间; C_{\max} 为所有工件的最大完工时间; X_{ijk} 为 0-1 二元变量,若工件 j 在阶段 i 被机器 k 加工,则其值为 1,否则为 0; Y_{ijt} 为 0-1 二元变量,若时刻 t 工件 j 正在阶段 i 上加工或阻塞,则其值为 1,否则为 0; Z_{ijl} 为 0-1 二元变量,若工件 j 在阶段 i 先于工件 l 加工,则其值为 1,否则为 0。

建立具有组合无限缓冲和阻塞约束的 MOHFESP-CBC 的整数规划模型如下所示。

$$\min C_{\max} = \max CT_{Sj}; \quad (1)$$

$$\min E = \sum_{i=1}^S \sum_{k=1}^{U_i} (E_{ik}^B + E_{ik}^W) = \sum_{k=1}^N \sum_{j=1}^N \sum_{i=1}^S V_{ik}^B P_{ijk} X_{ijk} + \sum_{i=1}^S \sum_{k=1}^{U_i} ((C_{ik}^{m_{ik}} - B_{ik}^1 - \sum_{j=1}^N P_{ijk} X_{ijk}) V_{ik}^W); \quad (2)$$

$$\sum_{k=1}^{U_i} X_{ijk} = 1, \forall i, j; \quad (3)$$

$$Z_{ijl} + Z_{ijl} \leq 1, \forall i, j \neq l; \quad (4)$$

$$BT_{il} + (3 - Z_{ijl} - X_{ijk} - X_{ilk})H \geq CT_{ij}, \forall i, k, j \neq l; \quad (5)$$

$$BT_{lj} \geq RT_j, \forall j; \quad (6)$$

$$\sum_{j=1}^N Y_{ijt} \leq U_i, \forall i, t; \quad (7)$$

$$\sum_{i=1}^T Y_{ijt} + TT_i = BT_{i+1,j} - BT_{ij}, \forall j, i < S, i \in W; \quad (8)$$

$$CT_{ij} + TT_i \leq BT_{i+1,j}, \forall j, i < S; \quad (9)$$

$$\sum_{i=1}^T Y_{ijt} = \sum_{k=1}^{U_i} X_{ijk} P_{ijk}, \forall j, i = S; \quad (10)$$

$$CT_{ij} = BT_{ij} + \sum_{k=1}^{U_i} X_{ijk} P_{ijk}, \forall i, j; \quad (11)$$

$$X_{ijk}, Y_{ijt}, Z_{ijl} \in \{0, 1\}, \forall i, j, l, k, t; \quad (12)$$

$$CT_{ij}, BT_{ij} \geq 0, \forall i, j. \quad (13)$$

其中,式(1)和式(2)表示需要优化的目标,即最小化最大完工时间和机器总能耗;式(3)表示任一工件在每个阶段仅能选取一台机器进行加工;式(4)和(5)保证了在同一台机器上的两个工件的加工非重叠性,即后一工件的开工时间不得小于前一工件的完工时间;式(6)表示每个工件在到达第一阶段之后才能开始加工;式(7)表示每个阶段任一时刻正在加工的工件数不能超过该阶段可利用的机器数;式(8)表示对于阻塞阶段集内的阶段 i ,工件占

用该阶段某台机器的时间段;式(9)表示只有工件完成当前阶段加工并运输到下一阶段后,才能开始后续阶段的加工;式(10)表示工件在最后一个阶段完工后即可离开;式(11)表示工件在加工过程中不能中断;式(12)和(13)定义了变量的取值范围。

1.3 多目标优化问题

解决多目标优化问题通常采用基于 Pareto 最优解的方法,一个多目标问题可以表示为

$$\min f(x) = \min \{f_1(x), f_2(x), \dots, f_i(x)\}。 \quad (14)$$

在多目标优化问题中,一般采用解的支配性来判断两个解的优劣。假设优化目标个数为 O 且都为最小化目标,则对于两个解 x_1 和 x_2 ,如果满足以下关系: $f_{l_1}(x_1) \leq f_{l_1}(x_2), \forall l_1 \in \{1, 2, \dots, O\}$ 且 $f_{l_2}(x_1) \leq f_{l_2}(x_2), \forall l_2 \in \{1, 2, \dots, O\}$, 则称 x_1 支配 x_2 , 当一个解不被其他任何解支配时,称其为 Pareto 最优解。所有 Pareto 最优解构成 Pareto 解集,在目标空间上的映射为 Pareto 前沿,而多目标优化问题的目标是尽可能得到更趋近于真实 Pareto 前沿的解集。

2 改进多目标模因算法

模因算法是进化算法的扩展,可以平衡全局搜索和局部搜索的优点。本文将 NSGA-II 融入全局优化算子,并设计一种融合自适应选择邻域搜索和多目标模拟退火的局部增强搜索算子,提出 IMOMA 来近似求解 MOHFESP-CBC。IMOMA 的主要过程如图 1 所示。

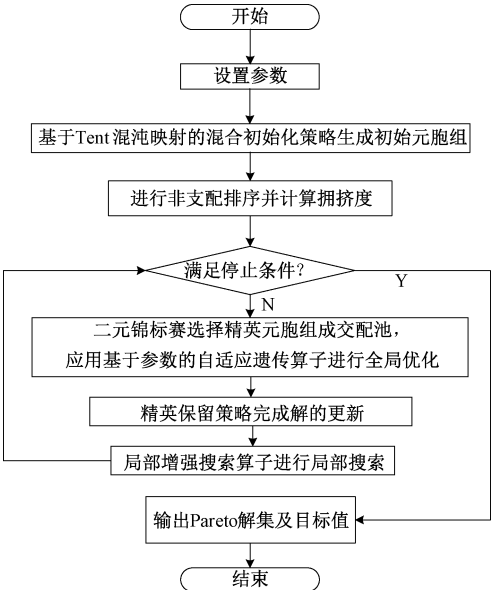


图 1 IMOMA 流程图
Figure 1 IMOMA flowchart

2.1 基于不相关机器分配的矩阵编码

不相关并行机环境下的混合流水线作业使得机器的分配对于完工时间和能耗的影响更大,因此采

用基于不相关机器分配的矩阵编码方案,构建包含工件加工序列和机器分配信息的 $N \times S$ 矩阵来表示每个可行解,行基因表示工件在各阶段加工的机器号,行顺序即为第一个生产阶段的工件加工序列,矩阵内元素 UM_{ji} 表示第 j 个位置的工件在阶段 i 所分配的机器号,如图 2 所示。

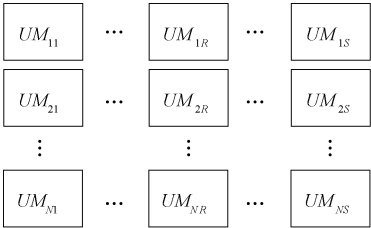


图 2 基于不相关机器分配的矩阵编码
Figure 2 Matrix coding based on uncorrelated machine assignment

2.2 解码

为获取调度解,根据矩阵内工件加工序列,依次将各工件安排在首个阶段所分配机器加工,令 $BT_{ij} \geq RT_j$ 以满足约束(6),后续阶段基于矩阵机器号信息为各工件分配机器,考虑约束(4)、(5)按先到先加工原则依次安排各工件的加工。对于无限缓冲阶段 i ,当工件 j 在阶段 i 的机器 k 上加工完成后,根据约束(3)和(7),若时刻 $CT_{ij} + TT_i$ 阶段 $i+1$ 上所分配机器 k' 空闲,则它进入阶段 $i+1$,否则工件 j 进入无限缓冲区,当机器 k' 完成其他工件加工后,累计机器 k' 的空闲时间段,根据约束(9)和(11)考虑加工和运输时间,当空闲时间段为 $P_{(i+1)jk'} - TT_i$ 时,工件 j 经运输后开始阶段 $i+1$ 的加工,同时机器 k' 的累计空闲时间段清零。同理,对于阻塞阶段 i ,若工件 j 在时刻 $CT_{ij} + TT_i$ 阶段 $i+1$ 上所分配机器 k' 空闲,则直接进入阶段 $i+1$ 加工,阻塞时间为 0,否则停留在当前机器 k 进行等待,记录机器 k' 的空闲时间段,直至空闲时间段为 $P_{(i+1)jk'} - TT_i$,工件 j 运输至阶段 $i+1$ 进行加工,根据约束(8)可得阻塞时间为 $BT_{(i+1)j} - (CT_{ij} + TT_i)$ 。下面以一个 $N=5, S=3, U_1=U_2=U_3=2$ 的算例来说明解码过程,其中仅考虑无限缓冲以及考虑阻塞阶段 $W=\{2\}$ 时的算例调度对比甘特图如图 3 所示。

图 3 体现了阻塞约束对于调度的影响,在考虑阶段 2 到阶段 3 之间有阻塞约束时,工件 4 在完成阶段 2 作业后,由于阶段 3 分配的机器 2 仍在加工,因此工件 4 阻塞在阶段 2 机器 2 上,直至经运输后阶段 3 的机器 2 空闲,工件 4 进入阶段 3,工件 5 开始阶段 2 的加工,阻塞时间为 2。对比仅考虑无限缓冲约束的调度甘特图,最大完工时间增加了 2。

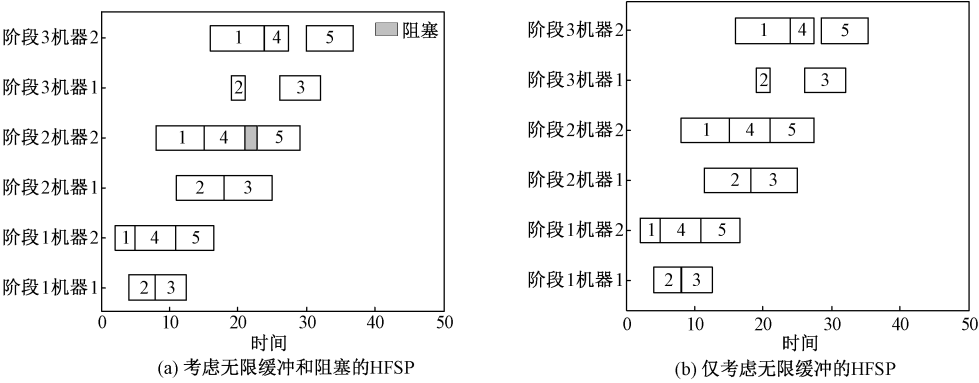


图 3 仅考虑无限缓冲与考虑阻塞阶段影响的调度对比

Figure 3 Comparison of scheduling between considering infinite buffering only and considering the impact of blocking

2.3 初始元胞数组

根据基于不相关机器分配的矩阵编码,每个矩阵解代表一个元胞个体, NP 个元胞个体构成元胞数组。高质量的初始元胞数组对算法的结果及全局收敛性有很大的影响。混沌映射具有随机性、遍历性及规律性等特点,相比经典的 Logistic 混沌映射, Tent 混沌映射得到的初始解具有更好的随机性、均匀性与多样性^[13],因此,采用基于 Tent 混沌映射的混合初始化策略生成初始元胞数组,其中由 Tent 混沌映射方法生成 $NP/2$ 数量的初始元胞数组,由随机均匀分布方法生成剩余 $NP/2$ 元胞数组,具体步骤如下。

步骤 1 随机生成 1 个初始混沌基因,按照式 (15) 混沌映射函数进行迭代,直至生成元胞的所有混沌基因。

$$Z_{c(h+1)} = \begin{cases} 2Z_{ch}, Z_{ch} < 0.5; \\ 2(1 - Z_{ch}), Z_{ch} \geq 0.5. \end{cases} \quad (15)$$

式中: c 为元胞序号; h 为混沌基因序号; $Z_{ch} \in [0, 1]$ 为混沌基因。

步骤 2 将 Z_{ch} 按式 (16) 转换到解空间得到对应值 X_{ch} ,对 X_{ch} 进行向下取整,即得到工件在各阶段所选的机器号。

$$X_{ch} = l_h + Z_{ch}(u_h - l_h). \quad (16)$$

式中: l_h 和 u_h 分别为混沌基因 h 对应工序可选择机器数量的下界和上界。

步骤 3 重复上述操作,直至生成 $NP/2$ 数量的元胞数组。

步骤 4 随机生成工件在各阶段加工机器号,考虑到机器使用的均衡性,令随机值服从 $[1, U_i]$ 的均匀分布,生成余下的 $NP/2$ 数量的元胞数组,共同构成最终的包含 NP 个元胞的初始元胞数组。

2.4 全局优化算子

2.4.1 选择策略

采用二元锦标赛选择方法选择父代元胞数组进入交配池进行交叉变异,过程如下。

步骤 1 通过快速非支配排序计算出元胞数组中各元胞的非支配等级和拥挤度。

步骤 2 从元胞数组中随机选择两个元胞进行比较,优先选择非支配等级高的元胞;若两者非支配等级相同,则优先选择拥挤度较高的元胞;若两者拥挤度相同,则从中任选一个。

步骤 3 重复步骤 2,直至选出 NP 个元胞作为父代元胞数组。

2.4.2 交叉变异算子

全局优化算子采用基于工件位的交叉算子和单点变异算子。

元胞数组通过交叉获得新的元胞,从而推动整个元胞数组向前进化,基于工件位的交叉算子是从父代元胞数组中随机选择两个元胞 $F1$ 与 $F2$,从中随机生成两个不重复的工件位 a 和 b 作为交叉点,交换 $F1$ 、 $F2$ 中 a 和 b 之间的所有行基因,从而得到两个新的子代元胞 $D1$ 和 $D2$,如图 4 所示。

变异操作通过随机对某些基因产生较小的扰动来生成新的元胞,采用单点变异算子,从父代元胞中随机选择一个工件位 γ 和阶段位 β ,为该位置基因所对应工序随机分配其他可用机器用以加工,如图 5 所示。

为使得算法在前期充分搜索,后期避免陷入局部最优同时保留优质信息,基于文献[14],设计一种基于参数的自适应遗传策略,使交叉概率随着迭代次数的增加而自适应降低,变异概率随着迭代次数的增加而自适应提高,交叉概率 PC 和变异概率 PM 分别为

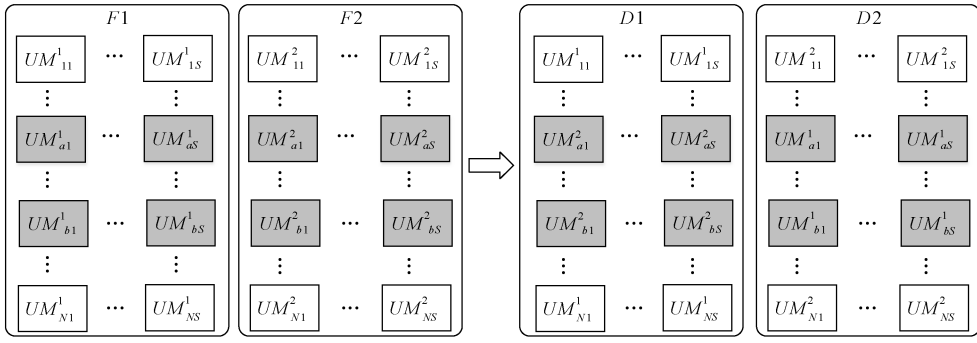


图 4 基于工件位的交叉算子

Figure 4 Crossover operator based on job position

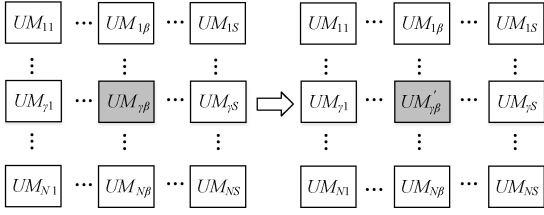


图 5 单点变异算子

Figure 5 Single-point mutation operator

$PC =$

$$\begin{cases} 0.05(G_1 - g)/G_1 + 0.9, & g \leq G_1; \\ 0.25(G_2 - g)/(G_2 - G_1) + 0.75, & G_1 < g \leq G_2; \\ 0.6(1 - \alpha)(G - g)/(G - G_2) + 0.6\alpha, & G_2 < g \leq G. \end{cases} \quad (17)$$

$$PM = 1 - PC. \quad (18)$$

式中: g 、 G 分别为当前迭代数和最大迭代数; G_1 为进化前期的最大迭代数,设为 $0.25G$; G_2 为进化中期的最大迭代数,设为 $0.75G$; α 为进化后期的调节系数,为避免后期 PC 值过小,设为 0.9 。

2.4.3 精英保留策略

精英保留策略通过将父代与子代元胞数组合并成规模为 $2NP$ 的新元胞数组,再根据非支配排序等级与拥挤度保留其中最优秀的前 NP 个元胞作为新一代元胞数组。

2.5 局部增强搜索算子

为提高搜索效率和解的质量,设计一种融合自适应选择邻域搜索和多目标模拟退火算法的局部增强搜索算子。

2.5.1 自适应选择邻域搜索

基于求解问题的特点设计 4 种邻域搜索结构来进行搜索产生新解,具体如下。

(1) N_1 : 插入。从当前元胞中随机选择两个工件行 a 和 b ($a \neq b$), 将行 a 插到行 b 之前。

(2) N_2 : 逆序反转。从当前元胞中随机选择两个工件行 a 和 b ($a \neq b$), 并反转两个行之间的基因。

(3) N_3 : 最大负载机器变异。根据解码策略计算当前元胞解中每个机器的能耗负载, 找到其中能耗负载最高的机器, 将分配到该机器的随机一个加工工件分配至该阶段其他随机机器上。

(4) N_4 : 最大负载机器重排。根据解码策略得到当前元胞解中能耗负载最高的机器, 随机选择两个分配至该机器加工的两个工件 a 和 b ($a \neq b$), 将该机器所在阶段的两工件 a 和 b 在序列中对应位置之间所有工件的加工机器重新分配。

为提高搜索效率, 采用自适应策略协调不同邻域结构之间的选择。每种邻域搜索结构具有一个概率值, 在执行完每次迭代的邻域搜索后更新概率值, 概率更新公式为

$$LP_u = np_u / \sum_{u=1}^4 np_u. \quad (19)$$

式中: LP_u 为邻域搜索结构 N_u 被选择的概率; np_u 为邻域搜索结构 N_u 的计数变量值。

在每次完成邻域搜索后, 假设当前执行的是 N_1 , 若得到的新解支配当前解, 则令 $np_1 = np_1 + 1$, 而 np_2 、 np_3 和 np_4 保持不变, 否则, 令 $np_u = np_u + 0.5$ ($u = 1, 2, 3, 4$)。为保证搜索多样性, 在每次迭代执行完邻域搜索后进行判定, 若某个邻域结构 N_u 的 $np_u > 200$, 则令 $np_u = np_u / 2$ ($u = 1, 2, 3, 4$)。

2.5.2 多目标模拟退火接受准则

完成每次的邻域搜索后, 需要判断是否接受当前解, 通常的接受准则为若新生成解更优, 即新生成解支配当前解, 则替换掉当前解。此操作虽简单易行, 但在多样性的保持上效果并不突出。因此, 使用基于多目标模拟退火的接受准则, 如果新解无法支配当前解, 仍然选择以一定概率接受当前解, 并且随着迭代后期温度的降低, 模拟退火对支配解的接受概率逐渐降低。基于多目标模拟退火的接受概率公式为

$$RP_g = \frac{1}{2} \sum_{m=1}^2 \exp\left(\frac{f_m^{\text{new}} - f_m^{\text{origin}}}{t_g}\right). \quad (20)$$

式中: RP_g 为第 g 次迭代的邻域搜索接受概率; t_g 为第 g 次迭代的温度值, $t_g = Temp \cdot t_{g-1}$, $Temp$ 为冷却因子。

3 实验设计与分析

3.1 实验设计

为检验 IMOMA 算法在求解 MOHFESP-CBC 的性能, 将其与传统 NSGA-II、结合局部搜索策略的改进混合蛙跳算法 ISLFA^[15] 以及结合改进精英保留策略和自适应交叉概率参数的改进 NSGA-II 算法 INSGA-II^[14] 进行对比。所有算法均采用 MATLAB R2014a 进行编程, 在处理器为 Intel Core i5-7300HQ CPU (2.50 GHz)、内存为 8.00 GB 的计算机上运行。为公平比较上述 4 种算法, 以 IMOMA 算法运行最大迭代次数 $Gen = 100$ 且最大运行时间不超过 600 s 为所耗时间作为其他 3 种对比算法的停止条件。测试算例数据如下: P_{ijk} 、 RT_j 、 TT_i 、 V_{ik}^B 和 V_{ik}^W 分别满足 $[1, 10]$ 、 $[1, 5]$ 、 $[1, 5]$ 、 $[10, 30]$ 和 $[1, 10]$ 的均匀分布。随机生成不同规模的算例进行仿真实验, 算例规模如下: $N \in \{20, 40, 60, 80, 100, 120\}$, $U_i \in \{4, 5\}$, $S \in \{3, 4\}$, 当 $S=3$ 时, $W=\{2\}$, 当 $S=4$ 时, $W=\{2, 3\}$ 。假设每个生产阶段的并行机数相同, $N \times S \times U_i$ 的不同组合共产生 24 种问题规模。

3.2 评价指标

多目标算法的优劣主要通过算法所求解集的非支配解数量、收敛性和多样性进行评估^[16]。本文选取反向世代距离指标 (IGD) 和间隔距离 (SP) 作为评价指标。由于测试算例的真实 Pareto 前沿未知, 因此使用近似 Pareto 前沿代替最优 Pareto 前沿, 对于每个算例, 将所有算法得到的非支配解集去除支配解和重复解可以得到该算例的近似 Pareto 前沿。

(1) IGD。IGD 是一种综合性指标, IGD 值越小, 表明解集越逼近近似 Pareto 前沿且分布更均匀, 解集的收敛性和多样性更好, 具体定义为

$$IGD = \frac{1}{|A|} \sum_{l=1}^{|A|} d_l. \quad (21)$$

式中: d_l 为第 l 个非支配解到最优解之间的欧氏距离; $|A|$ 为所得到的最优 Pareto 前沿中非劣解的个数。

(2) SP。SP 是一种多样性指标, 用来反映非支配解集中解分布的均匀性, SP 指标值越小, 解集中解的分布性与多样性越好, 具体定义为

$$SP = \sqrt{\frac{1}{|A| - 1} \sum_{l=1}^{|A|} (\bar{d} - d_l)^2}. \quad (22)$$

式中: \bar{d} 为所有 d_l 的均值。

3.3 不同规模问题对比实验

IMOMA 算法参数包括元胞数组规模 NP , 冷却因子 $Temp$ 。设定参数选取的大致范围: $NP \sim [80, 150]$, $Temp \sim [0.8, 0.99]$, 在模拟退火初始温度固定 10 000 的情况下, 在参数范围内进行大量算例测试, 以 IGD 值作为评价指标, 得到应用于 IMOMA 算法求解该问题的最优参数取值, 即 NP 为 100, 模拟退火初始温度为 10 000, $Temp$ 为 0.95。对比算法同样参考原文献设计取一定的参数取值范围进行测试, 从而设置 NSGA-II、ISLFA 和 INSGA-II 的 NP 均为 100; NSGA-II 中 $PC = 0.8$, $PM = 0.3$; ISLFA 中扰动变异循环次数为 5; INSGA-II 中进化后期阶段交叉率的调节系数为 0.8。

针对每种问题规模的算例, 每种算法独立运行 10 次, 每次运行得到一组 IGD 和 SP 值, 计算得到 10 次运行的平均 IGD 和平均 SP 值, 测试结果见表 1, 其中运行时间为求解对应规模算例时 4 种算法的运行时间。由表 1 可得到如下结论。

(1) 在 IGD 指标方面, 相同运行时间内 IMOMA 得到的平均 IGD 值在所有问题规模算例中均小于其他 3 种对比算法, 说明 IMOMA 在求解该问题时能得到更好的 Pareto 解集, 解的质量更优, 算法的寻优能力较强, 具有更好的收敛性和多样性。

(2) 在 SP 指标方面, 相同运行时间内 IMOMA 在求解 $40 \times 3 \times 4$ 和 $80 \times 4 \times 5$ 算例时获得的解集略差于 ISLFA, 但在其余算例上均优于其他 3 种对比算法, 说明 IMOMA 求得解集中的元胞解具有较好的分布性和多样性。

为探索算法运行时间与求解性能的关系, 取 $N = \{40, 60, 80, 100\}$, $S = 3$, $U_i = 5$ 的不同规模的一组算例, 分别运行 4 种算法, 设定停止条件为每种算法迭代 100 次, 独立运行 10 次, 得到 10 次的平均 IGD、平均 SP 值以及运行时间, 测试结果见表 2。

由表 2 可知, 对于 4 种规模问题, 在平均运行时间 199.8 s 内, IMOMA 得到的 IGD 和 SP 平均值均优于其他 3 种对比算法, 虽然在求解 $40 \times 3 \times 5$ 规模算例时 IMOMA 得到的 IGD 和 SP 值略差于 ISLFA, 但运行时间更短, 且随着问题规模的增大, IMOMA 得到的解集质量更高。IMOMA 的运行时间平均值短于 ISLFA, 略长于 NSGA-II 和 INSGA-II, 但在 IGD 和 SP 指标上的表现远远好于 NSGA-II 和 INSGA-II, 说明 IMOMA 用略长的时间得到了更高质量的解集。

表 1 不同问题规模算例测试结果

Table 1 Test results for different sized problem instances

$N \times S \times U_i$	IGD				SP				运行时间/s
	IMOMA	NSGA-II	ISLFA	INSGA-II	IMOMA	NSGA-II	ISLFA	INSGA-II	
20×3×4	0	1 525.20	1 362.10	1 335.00	338.00	2 059.10	1 827.00	1 673.00	67.59
20×3×5	0	1 941.10	1 553.00	2 174.40	109.00	2 050.00	1 709.00	3 267.00	63.86
20×4×4	84.51	3 010.90	332.43	3 046.80	255.20	3 496.30	583.20	3 566.90	74.08
20×4×5	0	3 166.30	524.61	2 720.60	129.80	3 443.70	653.62	3 170.80	76.33
40×3×4	346.12	1 620.60	629.72	1 306.50	1 857.20	4 731.00	1 693.20	3 698.80	175.34
40×3×5	0	2 798.70	3 060.10	2 695.10	589.00	3 399.00	3 815.00	3 284.00	179.53
40×4×4	4.55	4 595.30	412.63	4 889.10	851.20	5 987.10	1 256.00	6 417.70	167.45
40×4×5	0	6 889.70	1 485.30	6 816.50	376.00	8 059.90	1 861.20	7 435.30	156.41
60×3×4	295.20	5 705.10	2 574.10	5 873.30	1 208.40	6 831.60	2 896.80	6 940.80	197.96
60×3×5	0	5 100.60	1 789.30	5 150.30	1 398.00	7 815.80	2 948.00	7 571.60	178.03
60×4×4	0	5 179.40	440.39	3 749.40	1 722.40	7 401.00	2 122.40	5 590.30	210.42
60×4×5	0	8 026.90	1 725.10	7 949.50	644.00	9 019.80	2 369.00	9 333.40	219.66
80×3×4	0	5 897.90	792.28	6 974.10	672.00	6 842.50	1 464.20	8 379.30	248.65
80×3×5	83.31	6 432.50	1 303.10	5 881.90	343.21	7 806.80	1 649.00	6 497.40	242.19
80×4×4	129.86	8 097.90	489.29	7 308.60	1 305.40	9 784.90	1 504.00	8 897.70	264.23
80×4×5	46.67	6 887.10	68.26	6 741.90	1 587.00	8 738.60	1 468.30	8 707.60	272.95
100×3×4	3.65	7 029.00	1 152.30	6 979.40	1 051.80	8 748.60	2 171.20	8 596.10	294.78
100×3×5	151.43	8 767.70	1 762.40	7 282.60	1 070.00	10 121.00	2 379.40	8 415.70	291.75
100×4×4	0	8 272.50	1 866.10	7 792.80	1 139.00	10 042.00	2 990.20	9 589.40	348.25
100×4×5	0	9 717.10	1 952.10	9 717.10	1 075.30	12 090.00	3 027.40	11 410.00	371.39
120×3×4	4.08	7 495.20	1 443.70	8 383.00	867.67	8 550.50	2 311.40	9 972.50	400.36
120×3×5	0	8 936.30	2 326.00	8 649.40	1 027.00	10 799.00	3 353.00	10 843.00	374.50
120×4×4	0	12 075.00	1 688.30	11 399.00	419.00	12 779.00	2 107.10	13 171.00	455.52
120×4×5	0	10 088.00	3 124.50	10 733.00	538.33	11 131.00	3 662.80	11 658.00	458.98
平均	47.89	6 219.00	1 410.71	6 064.55	857.25	7 572.01	2 159.27	7 420.30	241.26

表 2 各算法迭代 100 次的测试结果

Table 2 Test results of algorithms for 100 iterations

$N \times S \times U_i$	IGD				SP				运行时间/s			
	IMOMA	NSGA-II	ISLFA	INSGA-II	IMOMA	NSGA-II	ISLFA	INSGA-II	IMOMA	NSGA-II	ISLFA	INSGA-II
40×3×5	106.1	3 401.5	82.0	3 556.2	250.0	3 642.4	165.3	3 885.4	136.6	92.5	171.8	104.0
60×3×5	0	5 417.1	222.0	5 156.1	63.0	5 705.8	285.0	5 279.5	173.0	138.0	229.2	137.9
80×3×5	18.5	7 154.2	1 074.6	6 990.2	868.3	8 877.1	1 942.7	8 296.8	209.6	160.9	251.3	162.8
100×3×5	36.9	9 604.7	2 088.5	9 284.7	721.3	10 803.2	3 009.8	10 604.0	280.0	193.9	343.9	203.5
平均	40.4	6 394.4	866.8	6 246.8	475.7	7 257.1	1 350.7	7 016.4	199.8	146.3	249.1	152.0

为判断算法之间是否存在统计学意义上的显著差别,根据表 2 的不同算例测试数据,使用 SPSS Statistics 21 进行 Wilcoxon 符号秩检验,结果如表 3 所示。由表 3 可知,在 5% 的显著性水平下, P 值均 < 0.5,因此对于 IGD 和 SP 指标,IMOMA 算法对比其余 3 种算法具有压倒性的显著优势。

为更直观表现算法求解效果,以测试问题 80×3×5 为例,4 种算法各随机运行 1 次获得的 Pareto 前沿如图 6 所示,其中,“APF”表示该问题的近似 Pareto 前沿。由图 6 可知,在 APF 的 6 个解中,有 5 个解由 IMOMA 得到,1 个解由 ISLFA 得到,而 NSGA-II 和 INSGA-II 得到的 Pareto 解集质量相对

较差。

表 3 Wilcoxon 符号秩检验结果

Table 3 Wilcoxon sign rank test results

算法比较	P 值	
	IGD	SP
IMOMA 和 NSGA-II	0.000 018	0.000 018
IMOMA 和 ISLFA	0.000 018	0.000 027
IMOMA 和 INSGA-II	0.000 018	0.000 018

同时,相较于其他 3 种对比算法,IMOMA 所得解集更接近于近似 Pareto 前沿,具有更好的收敛性。因此,IMOMA 算法在求解 MOHFESP-CBC 方面具有更好的算法性能。

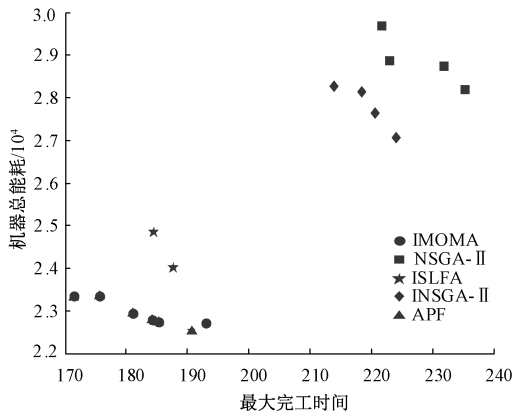


图 6 4 种算法求解 80×3×5 算例的 Pareto 前沿

Figure 6 Pareto front of 4 algorithms for solving the 80×3×5 case

4 结论

本文研究了生产阶段间带有无限缓冲和阻塞两种不同中间缓冲约束的 HFSP,以最小化最大完工时间和机器总能耗为目标建立了 MOHFESP-CBC 的整数规划模型。提出一种 IMOMA 算法求解该问题,应用基于 Tent 混沌映射的混合初始化策略生成初始元胞数组,全局优化算子引入 NSGA-II,采用基于参数的自适应遗传策略对交叉概率、变异概率进行调整,局部增强搜索算子采用融合自适应选择邻域搜索和多目标模拟退火的搜索策略,并结合问题特点设计 4 种邻域搜索结构用于提升局部搜索效率。

通过不同问题规模的仿真实验,与 3 种现有算法进行对比,验证了 IMOMA 在求解 MOHFESP-CBC 算法性能的有效性和优越性。

节能调度是现代工业发展的一大趋势,未来可继续深入探讨节能和经济指标的综合在车间调度领域的应用研究;还可考虑加入更多的实际生产约束以及其他组合中间缓冲约束的 HFSP;另外探索更优的编解码方式以及设计解决该类问题的更高效的智能算法也将是下一步研究的重点。

参考文献:

[1] RIAHI V, KHORRAMIZADEH M, HAKIM NEWTON M A, et al. Scatter search for mixed blocking flowshop scheduling[J]. Expert Systems with Applications, 2017, 79: 20-32.

[2] 王君妍,王薛苑,轩华. 带批处理机的多阶段柔性流水车间调度优化[J]. 郑州大学学报(工学版), 2017, 38(5): 86-90.

WANG J Y, WANG X Y, XUAN H. Multi-stage flexible flowshop scheduling with batching machines[J]. Journal

of Zhengzhou University (Engineering Science), 2017, 38(5): 86-90.

[3] QIN H X, HAN Y Y, WANG Y T, et al. Intelligent optimization under blocking constraints: a novel iterated greedy algorithm for the hybrid flow shop group scheduling problem[J]. Knowledge-Based Systems, 2022, 258: 109962.

[4] MISSAOUI A, BOUJELBENE Y. An effective iterated greedy algorithm for blocking hybrid flow shop problem with due date window[J]. RAIRO-Operations Research, 2021, 55(3): 1603-1616.

[5] 胡蓉,董钰明,钱斌. 基于探路者算法的绿色有限缓冲流水线调度[J]. 系统仿真学报, 2021, 33(6): 1384-1396.

HU R, DONG Y M, QIAN B. Pathfinder algorithm for green pipeline scheduling with limited buffers[J]. Journal of System Simulation, 2021, 33(6): 1384-1396.

[6] LIN C C, LIU W Y, CHEN Y H. Considering stockers in reentrant hybrid flow shop scheduling with limited buffer capacity[J]. Computers & Industrial Engineering, 2020, 139: 106154.

[7] 裴小兵,李依臻. 新型混合改进遗传算法求解零等待流水车间调度问题[J]. 计算机集成制造系统, 2021, 27(3): 815-827.

PEI X B, LI Y Z. New hybrid improved genetic algorithm for solving no-wait flow shop scheduling problem[J]. Computer Integrated Manufacturing Systems, 2021, 27(3): 815-827.

[8] AZAIEZ M N, GHARBI A, KACEM I, et al. Two-stage no-wait hybrid flow shop with inter-stage flexibility for operating room scheduling[J]. Computers & Industrial Engineering, 2022, 168: 108040.

[9] WANG Y M, LI X P, RUIZ R, et al. An iterated greedy heuristic for mixed no-wait flowshop problems[J]. IEEE Transactions on Cybernetics, 2018, 48(5): 1553-1566.

[10] 钟祯充,李文锋,贺利军,等. 集装箱码头混合零空闲柔性流水作业调度优化[J]. 计算机集成制造系统, 2022, 28(11): 3421-3432.

ZHONG L C, LI W F, HE L J, et al. Optimization of mixed no-idle flexible flow scheduling in container terminal[J]. Computer Integrated Manufacturing Systems, 2022, 28(11): 3421-3432.

[11] 轩华,付鑫博,李冰. 基于混合离散人工蜂群算法的混合零等待柔性流水车间优化研究[J]. 工业工程与管理, 2023, 28(1): 170-180.

XUAN H, FU X B, LI B. Research on optimization of mixed zero-wait flexible flowshop based on hybrid discrete artificial bee colony algorithm[J]. Industrial Engineering and Management, 2023, 28(1): 170-180.

[12] ZHUANG Z L, ZHANG Z L, TENG H, et al. Optimiza-

tion for integrated scheduling of intelligent handling equipment with bidirectional flows and limited buffers at automated container terminals[J]. Computers and Operations Research, 2022, 145: 105863.

[13] 姜一啸, 吉卫喜, 何鑫, 等. 基于改进非支配排序遗传算法的多目标柔性作业车间低碳调度[J]. 中国机械工程, 2022, 33(21): 2564–2577.

JIANG Y X, JI W X, HE X, et al. Low-carbon scheduling of multi-objective flexible job-shop based on improved NSGA-II [J]. China Mechanical Engineering, 2022, 33 (21): 2564–2577.

[14] 黄辉, 李梦想, 严永. 考虑序列设置时间的混合流水线车间多目标调度研究[J]. 运筹与管理, 2020, 29 (12): 215–221.

HUANG H, LI M X, YAN Y. Research on multi-objective scheduling of hybrid flow production shop considering sequence setting time[J]. Operations Research and Management Science, 2020, 29(12): 215–221.

[15] 张洪亮, 徐公杰, 张金春. 带有学习效应的多阶段混合流水车间调度问题研究[J]. 重庆师范大学学报(自然科学版), 2021, 38(1): 87–95.

ZHANG H L, XU G J, ZHANG J C. Research on multi-stage hybrid flow shop scheduling problem with learning effect[J]. Journal of Chongqing Normal University (Natural Science), 2021, 38(1): 87–95.

[16] 王丽萍, 任宇, 邱启仓, 等. 多目标进化算法性能评价指标研究综述[J]. 计算机学报, 2021, 44(8): 1590–1619.

WANG L P, REN Y, QIU Q C, et al. Survey on performance indicators for multi-objective evolutionary algorithms [J]. Chinese Journal of Computers, 2021, 44 (8): 1590–1619.

Multi-objective Hybrid Flowline Energy-saving Scheduling with Combined Buffer Constraints

XUAN Hua, GENG Zhuxin, LI Bing

(School of Management, Zhengzhou University, Zhengzhou 450001, China)

Abstract: To solve the hybrid flowline energy-saving scheduling problem with two intermediate buffer constraints, infinite buffer and blocking, between production stages, a mathematical model was formulated by considering the uncorrelated parallel machines and multiple time constraints. Taking into account the characteristics of the problem, an improved multi-objective memetic algorithm (IMOMA) was proposed to minimize simultaneously makespan and total energy consumption of the machines. The algorithm adopted a matrix encoding method based on uncorrelated machine assignment. Using a hybrid initialization strategy based on Tent chaotic map to generate the initial cell array, an non-dominated sorting genetic algorithm improved by parameter-based adaptive genetic strategy was applied for the global optimization operator, and a search strategy integrating adaptive selection neighborhood search and multi-objective simulated annealing was designed for the locally enhanced search operator to improve the algorithm’s search capability. The effectiveness and superiority of the proposed algorithm were verified through case experiments with 24 problem scales. The experimental results showed that the average *IGD* value of 47.89 and the average *SP* value of 857.25 obtained by IMOMA within the average running time of 241.26 s were lower than the other three comparison algorithms. So the solution set obtained by IMOMA had better convergence, diversity and distributivity.

Keywords: hybrid flowline; improved multi-objective memetic algorithm; combined buffer constraints; unrelated parallel machine; multi-objective optimization; energy-saving scheduling