

系统仿真学报
Journal of System Simulation
ISSN 1004-731X, CN 11-3092/V

《系统仿真学报》网络首发论文

题目: 基于混合离散状态转移算法的多目标节能无等待流水车间调度
作者: 王聪, 余佳英, 张宏立
DOI: 10.16182/j.issn1004731x.joss.23-0743
收稿日期: 2023-06-19
网络首发日期: 2023-10-10
引用格式: 王聪, 余佳英, 张宏立. 基于混合离散状态转移算法的多目标节能无等待流水车间调度[J/OL]. 系统仿真学报.
<https://doi.org/10.16182/j.issn1004731x.joss.23-0743>



网络首发: 在编辑部工作流程中, 稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定, 且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式(包括网络呈现版式)排版后的稿件, 可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定; 学术研究成果具有创新性、科学性和先进性, 符合编辑部对刊文的录用要求, 不存在学术不端行为及其他侵权行为; 稿件内容应基本符合国家有关书刊编辑、出版的技术标准, 正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性, 录用定稿一经发布, 不得修改论文题目、作者、机构名称和学术内容, 只可基于编辑规范进行少量文字的修改。

出版确认: 纸质期刊编辑部通过与《中国学术期刊(光盘版)》电子杂志社有限公司签约, 在《中国学术期刊(网络版)》出版传播平台上创办与纸质期刊内容一致的网络版, 以单篇或整期出版形式, 在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊(网络版)》是国家新闻出版广电总局批准的网络连续型出版物(ISSN 2096-4188, CN 11-6037/Z), 所以签约期刊的网络版上网络首发论文视为正式出版。

基于混合离散状态转移算法的多目标节能无等待流水车间调度

王聪, 余佳英, 张宏立*

(新疆大学 电气工程学院, 新疆 乌鲁木齐 830047)

摘要: 针对以完工时间和总能耗为目标的节能无等待流水车间调度问题(EENWFSP), 设计一种混合离散状态转移算法(HDSTA)进行求解。根据问题特性设计工序和速度矩阵的编码方式, 采用启发式算法获得优质的初始解。根据 EENWFSP 性质, 对四个离散操作算子进行任务分配, 将嵌入二次状态转移的交换、移动、对称算子用于工序优化, 替换算子用于机器速度优化, 并在替换算子中嵌入基于关键路径的速度替换策略。最后设计了一种改进的破坏重构操作用于进一步提高 Pareto 解的质量。通过与 4 种算法在测试实例上的实验结果分析, 表明 HDSTA 在解决 EENWFSP 时具有较强优势。

关键词: 无等待流水车间; 节能调度; 状态转移算法; 总能量消耗; 最大完工时间; 多目标优化

中图分类号: TP301.6; TP391.9

文献标志码: A

DOI: 10.16182/j.issn1004731x.joss.23-0743

Multi-objective energy-efficient no-wait flow shop scheduling based on hybrid discrete state transition algorithm

Wang Cong, Yu Jiaying, Zhang Hongli*

(School of Electrical Engineering, Xinjiang University, Urumqi 830047, China)

Abstract: A hybrid discrete state transition algorithm (HDSTA) is designed to solve the energy-efficient no-wait flow shop scheduling problem (EENWFSP) with the goal of makespan and total energy consumption. According to the characteristics of the problem, the coding method of job sequence and speed matrix is designed, and the heuristic algorithm is used to obtain the high-quality initial solution. According to the properties of EENWFSP, four discrete operators are assigned tasks. The swap, shift and symmetry operators embedded with secondary state transition are used for job sequence optimization, and the substitute operators are used for machine speed optimization. The speed substitute strategy based on critical path is embedded in the substitute operators. Finally, an improved destructive reconstruction operation is designed to further improve the quality of Pareto solution. Through the analysis of the experimental results with four algorithms on test cases, it shows that HDSTA has strong advantages in solving EENWFSP.

收稿日期: 2023-06-19

修回日期: 2023-08-08

基金项目: 自治区自然科学基金(2022D01C367), 自治区杰青项目(2022D01E33), 国家自然科学基金(52267010), 天池博士计划(TCBS202115)。

第一作者: 王聪 (1989-), 女, 讲师, 博士, 从事智能控制、群智能算法的研究, E-mail: 641087385@qq.com;.....

通讯作者: 张宏立(1972-), 男, 教授, 博士, 研究方向为复杂生产过程优化与调度等。E-mail: zh1xju@163.com.

Key words: no-wait flow shop scheduling problem; energy-efficient scheduling; state transition algorithm; total energy consumption; makespan; multi-objective optimization

0 引言

2020 年我国提出“碳达峰、碳中和”目标以来,碳排放已经成为制造企业生产中日益重视的指标。近年来,针对节能车间调度的研究受到大量学者的关注^[1-2]。目前,教与学算法^[3],布谷鸟算法^[4],候鸟优化算法^[5],蛙跳算法^[6],粒子群算法^[7]等群智能算法已成功应用于节能车间调度问题中。

无等待流水车间调度问题(No Wait Flow Shop Scheduling Problem, NWFSP)是离散工业和流程工业中常见的优化问题之一,由于其具有较强的理论深度以及较广的应用背景,近年来得到了广大学者的研究^[8-9]。已经证明,当机器数目大于 2 时, NWFSP 是 NP-hard 问题^[10],然而大多数学者解决 NWFSP 问题时,只考虑完工时间和成本等经济指标,并未考虑碳排放等能耗指标。因此本文考虑解决节能无等待流水车间调度问题(Energy Efficient No Wait Flow Shop Scheduling Problem, EENWFSP)。

目前对于带能源指标的流水车间调度主要集中于设定机器的加工速度可调,即机器处理速度越大,工件加工过程耗时越短但能源消耗越高^[11]。EENWFSP 通常包含两个子问题:1) 工序调度子问题,即工件排序问题;2) 机器速度选择子问题,即确定各工序加工时对应机器的速度。两个子问题相互影响,相互制约,相较于传统的流水车间调度问题更为复杂。因此,应该确定合适的工件排序和适中的机器加工速度,以取得生产效率和能源效率的标准之间的权衡。近年来,具有动态速度的节能调度问题得到了广泛的研究。例如:Liu^[12]等针对最小化总加权拖期,总碳排放和峰值能耗为目标的批处理流水线调度问题,提出了一种自适应多目标遗传算法,实验结果表明该方法优于传统多目标遗传算法。Ding^[13]等分析节能流水车间调度问题两个目标的特点,提出融合节能策略的多目标

Nawaz-Enscore-Ham(NEH)算法以及改进的多目标迭代贪婪算法用于求解低碳流水线调度问题。Jiang^[14]等提出了一种基于分解的改进多目标进化算法求解带有序列相关准备时间低碳流水车间调度问题。Tang^[7]等提出一种改进的粒子群算法解决节能柔性流水车间调度问题。Tasgetiren^[15]等人结合破坏重构操作以及迭代搜索算法重新设计离散人工蜂群算法用于求解双目标 NWFSP 问题。Wang^[16]等人基于节能分布式流水车间特点设计一种基于特定知识的协同进化算法,用于最小化完工时间和总能耗。艾子义^[6]等人采取基于种群和记忆的种群划分方法,提出了一种新型蛙跳算法用于求解低碳柔性作业车间调度问题。潘子肖^[17]等人针对分布式低碳并行机调度问题,提出了一种基于问题性质的非劣排序遗传算法用于优化延迟时间和总能耗。仿真结果表明,该方法求解该问题具有较强优势。上述算法在求解低碳车间调度问题时取得了较好的效果,但上述研究少有考虑到无等待约束条件,随着无等待制造不断应用于各个行业,有必要开展对 EENWFSP 的深入研究。虽然 NWFSP 已得到广泛的研究,在加入能耗约束后,需要同时优化完工时间以及总能耗两个目标,这对问题的求解带来了额外的困难,例如,在进行机器加工速度调整时,会破坏原有的无等待约束,基础的低碳流水车间机器速度调整方法在 EENWFSP 上并不适用。

状态转移算法(State Transition Algorithm, STA)是 2012 年由 Zhou^[18]等提出的一种新型结构主义智能算法,近几年得到了广泛的应用。阳春华^[19]等人提出了一种离散状态转移算法(Discrete State Transition Algorithm, DSTA)用于求解旅行商问题,仿真结果表明,该算法具有耗时短、寻优能力强等优点。Zhou^[20]等人分析了基于“恢复概率”和“风险概率”的参数研究,提出了一种新型 DSTA 用于求解配水网络优化问题。董天雪^[21]等人在一次状态

转移的基础上提出了二次状态转移的概念,融合停滞回溯策略,提出了一种改进 DSTA,仿真结果表明,该方法在求解员工指派问题上优于模拟退火算法。这些学者的工作扩展了 STA 的优势,拓宽了算法的优化思路。由于 DSTA 具有全局性、最优性、快速性和可控性等显著特点,且四种离散状态转移算子可分别完成交换、移动、对称和替换功能,比起其它优化算法更适用于 EENWFSP 的求解,因此本文选择 DSTA 对 EENWFSP 进行求解,但基本离散状态转移算法在求解多目标 EENWFSP 时难以实现深度搜索,因此本文从 EENWFSP 的基本性质出发,结合离散操作,提出一种多目标混合离散状态转移算法 (Hybrid Discrete State Transition Algorithm, HDSTA)。

本文以最小化总能耗和完工时间为目标,建立相应的 EENWFSP 的数学模型,首先,利用 NN(Nearest Neighbor)和以工件加工时间标准差为优先级的 NEH 启发式算法获得初始工件调度方案,然后合理分配 DSTA 中各离散状态转移算子,将嵌入二次状态转移操作的交换、移动和对称算子用于工序优化阶段,将基于关键路径的替换算子用于速度优化阶段,在算法最后加入基于破坏重构的混合优化阶段用于提升非支配解质量。通过仿真结果验证了 HDSTA 改进策略的有效性,相较四种对比算法,HDSTA 获得的非支配解数目更多,收敛性更好且分布也更均匀。

1 节能无等待流水车间调度问题

1.1 符号定义

本文中问题描述中所使用的符号含义如表 1 所示:

表 1 模型相关参数

Table 1 Relevant parameters of the model

模型参数	参数含义
N	工件总数
M	机器总数
$p_{j,i}$	工件 j 在机器 i 上的标准加工时间
v_l	速度等级 l 的速度系数, $l \in L$, $L = \{1, 2, 3\}$

φ_i	处于待机模式的机器 i 单位时间的能量消耗, $i \in M$
$\tau_{i,l}$	机器 i 以速度 l 运行时单位时间的能量消耗, $i \in M$
D	一个极大正数
$y_{j,i,l}$	0~1 变量,若机器 i 上以速度 l 加工工件 j 时为 1, 否则为 0
$x_{j,k}$	0~1 变量,若工件 j 在工件 k 之前加工时为 1, 否则为 0
$C_{j,i}$	工件 j 在机器 i 的完工时间
θ_i	机器 i 上的空闲时间
C_{\max}	最大完工时间
TEC	总能源消耗

1.2 问题描述

EENWFSP 调度问题的数学描述为: N 个工件 ($j = 1, 2, \dots, N$) 按照相同的加工次序依次在 M 台机器 ($i = 1, 2, \dots, M$) 上加工,各工件加工时无等待时间。考虑到机器速度缩放操作, EENWFSP 中每台机器均有三档加工速度可以调整,因此各工序的加工时间会随着机器速度的改变而变动。设定速度集合为 $L = \{1, 2, 3\}$, 为所有机器可选择的速度档数,各档对应的真实速度集合为 $v = \{v_1, v_2, v_3\}$, 各档对应单位时间内的能源消耗为 $\tau = \{\tau_1, \tau_2, \tau_3\}$ 。并存在以下约束: (1) 各机器在任一时刻至多只能加工一个工件; (2) 各工件在同一时刻只能被一台机器加工; (3) 为满足无等待约束,除了第一台机器上第一个加工工件之外,其余所有工序的开工时间都被推迟; (4) 机器在工件加工过程中速度不能改变。EENWFSP 相较于传统的 NWFSP 调度问题来说,求解更为复杂,需要确定工件的加工顺序和各机器的加工速度,使最大完工时间 C_{\max} 以及机器总能耗 (Total Energy Consumption, TEC) 最小,其数学模型如下:

$$\text{Minimize}(C_{\max}) = \min(C_{n,m}) \quad (1)$$

$$\text{Minimize}(TEC) \quad (2)$$

$$s.t. \quad C_{j,l} \geq \sum_{i \in L} \frac{p_{j,i} y_{j,i,l}}{v_l} \quad (3)$$

$$C_{j,i} - C_{j,i-1} \geq \sum_{l \in L} \frac{p_{j,i} y_{j,i,l}}{v_l} \quad \forall j \in N, \forall i \in M : i \geq 2 \quad (4)$$

$$C_{j,i} - C_{k,i} + Dx_{j,k} \geq \sum_{l \in L} \frac{p_{j,i} y_{j,i,l}}{v_l} \quad \forall j, k \in N : k \neq j, \forall i \in M \quad (5)$$

$$C_{j,i} - C_{k,i} + Dx_{j,k} \leq D - \sum_{l \in L} \frac{p_{j,i} y_{j,i,l}}{v_l} \quad \forall j, k \in N : k \neq j, \forall i \in M \quad (6)$$

$$C_{\max} \geq C_{j,m} \quad \forall j \in N \quad (7)$$

$$C_{j,i} - C_{j,i-1} \leq \sum_{l \in L} \frac{p_{j,i} y_{j,i,l}}{v_l} \quad \forall j \in N, \forall i \in M, i \geq 2 \quad (8)$$

$$\sum_{l \in L} y_{j,i,l} = 1; \quad \forall j \in N, \forall i \in M \quad (9)$$

$$TEC = \sum_{j \in N} \sum_{i \in M} \sum_{l \in L} \frac{p_{j,i} \tau_{i,l}}{v_l} y_{j,i,l} + \sum_{i \in M} \varphi_i \theta_i \quad (10)$$

$$\theta_i = C_{\max} - \sum_{j \in N} \sum_{l \in L} \frac{p_{j,i} y_{j,i,l}}{v_l} \quad \forall i \in M \quad (11)$$

$$y_{j,i,l} \in \{0,1\}, C_{j,i} \geq 0 \quad \forall j \in N, \forall i \in M, \forall L_s \in L, \\ x_{j,k} \in \{0,1\} \quad \forall j, k \in N, k \neq j \quad (12)$$

式(1)表示最小化完工时间；式(2)表示最小化机器总能耗；式(3)表示在第一台机器上加工的工件，其完工时间不小于其加工时间；式(4)表示工件 j 只有在完成了在机器 $i-1$ 上的加工之后才能在机器 i 上加工；式(5)和式(6)保证工件 j 和工件 k 在调度序列中是单一出现的；式(7)用于计算最大完工时间；式(8)和(9)保证工件 j 在机器 i 上仅以一种速度加工，且结合约束(4)可得：当 $i \geq 2$ 时，工件 j 的各子工序的开始时间等于上一道工序的结束时间，即加工过程满足无等待约束；式(10)用于计算总能源消耗；式(11)用于计算每台机器的空闲时间；式(12)定义了所有决策变量。

2 离散状态转移算法

STA 是近几年发展起来的一种新型全局优化算法，它将优化问题的可行解看作状态，将解更新的过程看作状态转移过程，基本 DSTA 形式如下：

$$\begin{cases} x_{k+1} = G_k x_k \\ y_{k+1} = f(x_{k+1}) \end{cases} \quad (13)$$

式中， $x_k = [x_{1k}, x_{2k}, \dots, x_{nk}]^T$ ， $x_{ik} \in \{1, 2, \dots, n\}$ 对应优化问题的解； $G_k \in R^{n \times n}$ 是状态转移矩阵，是每行每

列只有一个位置等于 1 的一般初等矩阵； $f(x)$ 是评价函数。基本 DSTA 是基于个体的随机搜索技术，它从当前最优解出发，在离散状态算子的操作下，产生具有算子同性质的邻域，然后从该邻域中抽样大小为 SE 的状态集，并采用贪婪策略更新当前最优解。从限于篇幅限制，DSTA 的原理和步骤参考文献[18]。

3 混合离散状态转移算法求解 EENWFSP

对于 DSTA，各学者主要是集中于单目标优化问题的研究，本文在基本 DSTA 的基础上，结合 EENWFSP 调度问题性质，提出一种基于 Pareto 最优的多目标 HDSTA 用于求解该问题。为解决状态与调度方案的一一对应关系，分别设置了工件排序与速度矩阵的编码方式。为了提升初始解的质量，使用基于 NEH 的启发式方法以及适当的速度分配方式进行初始化。DSTA 的核心在于状态更新，而该算法采用基于最优解的单点搜索方法，更新策略需要通过找到每一代的最优解才能实现，但 EENWFSP 为多目标调度问题，不能直接得到最优解。因此本文将采样产生的所有解进行非支配排序，并利用外部档案集(Archives Set, AS)进行储存。此外，将交换，移动和对称算子加入二次状态转移操作，并对速度替换算子加入基于关键路径的替换操作。最后增加改进的破坏重构混合优化阶段进一步提高 Pareto 解的质量。

3.1 个体编码方式

为了利用 DSTA 求解 EENWFSP 调度问题，一是需要建立工序矩阵解与向量排列之间的一一对应关系，二是建立速度矩阵解与向量排列之间的一一对应关系。首先，当 $N=5$ ，即工件数目为 5，满足工序约束条件的变量 x_{ik} 组成的可行解可以用式(14)中的矩阵 M_{array} 表示(i 表示机器， k 表示工件)。其次，当 $N=5$ ， $M=5$ ，满足速度约束条件的变量 v_{ik} 组成的可行解可以用式(15)中的矩阵 M_{SLM} 表示，假设 $\max(L)=3$ 时，式(15)中 $v_{12}=2$ 表示工件二在第

一台机器上加工时, 机器一的加工速度为等级 2。

$$M_{\text{array}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Leftrightarrow \begin{pmatrix} x_{11} = 1 \\ x_{24} = 1 \\ x_{33} = 1 \\ x_{42} = 1 \\ x_{55} = 1 \end{pmatrix} \Leftrightarrow \begin{pmatrix} \pi(1) = 1 \\ \pi(2) = 4 \\ \pi(3) = 3 \\ \pi(4) = 2 \\ \pi(5) = 5 \end{pmatrix} \quad (14)$$

$$M_{\text{SLM}} = \begin{bmatrix} V(1,1) & V(1,2) & \dots & V(1,n) \\ V(2,1) & V(2,2) & \dots & V(2,n) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ V(m,1) & V(m,2) & \dots & V(m,n) \end{bmatrix} \Leftrightarrow \begin{bmatrix} 1 & 2 & 1 & 3 & 1 \\ 2 & 1 & 3 & 1 & 2 \\ 1 & 3 & 1 & 3 & 2 \\ 2 & 1 & 3 & 1 & 1 \\ 1 & 2 & 2 & 1 & 1 \end{bmatrix} \quad (15)$$

采用上述编码方式优势在于经过状态算子操作后的解依然可行, 避免了算法在求解过程中对非法解的修正, 缩短了调度问题的求解时间。

3.2 初始化方法

EENWFSP 调度问题需要初始化工件的排序以及机器的加工速度。对于工件排序的初始化, 分析该问题可知, 当机器速度等级固定时, C_{\max} 越小, TEC 越小。因此本文在初始化阶段采用 NN(Nearest Neighbor)和以工件加工时间标准差为优先级的 NEH 启发式算法^[9]。步骤如下:

Step1: 从工件集合 $j=1,2,\dots,N$ 中每次随机选择 1 个工件, 执行 NP 次, 这些工件组成初始种群 $JF_s = \{JF_1, JF_2, \dots, J_{NP}\}$ 中的第一个工件 $\pi_i(1)$;

Step2: 利用 NN 启发式规则组成 $\pi_i(2)$;

Step3: 使用以工件加工时间标准差为优先级的 NEH 方法决定剩余 $(N-2)$ 个工件的顺序;

Step4: 以初始序列 JF_s 重复 step2 和 step3 形成初始工件排序种群。

工件排序初始化完成后, 对初始工件种群进行适应度值计算 (此时各机器处理速度均为中等速度), 将适应度最优的工件排序记作 π_{best} , 再设置速度矩阵初始化规模为 NP , 为 π_{best} 所有工件分别分配最高、最低和中等速度, 将其构建为初始种群的第一、第二和第三个个体, 为了增加初始解的多

样性, 速度种群中剩余个体由为 π_{best} 随机分配速度等级构成。

3.3 外部档案集的构建、更新和筛选

在多目标优化问题中, 一般情况下无法求得最优解, 往往是在求解过程进行取舍求得一组 Pareto 解集。为避免算法在迭代过程中遗失优良个体, 本文建立一个外部档案集 AS。然后对 3.2 节初始种群进行非支配排序计算, 将该种群的非支配解填入 AS 作为初始外部档案集。

更新准则是基于 Pareto 准则更新, 具体操作如下: 若更新后种群中的解 x^* 支配 AS 中所有解, 则直接将其加入 AS 集; 若更新后种群中的解 x^* 支配 AS 中部分解, 则将其加入 AS 集, 并将 AS 中受 x^* 支配的解剔除。为控制计算复杂度, 将 AS 长度设置为 K , 在算法更新过程中, 若 AS 的长度超过了 K , 则需要删除 AS 中拥挤距离更小的解, 以控制外部档案集的规模。

3.4 EENWFSP 的性质

在分析 EENWFSP 性质之前给出以下定义:

定义 1: (1) 关键路径: 对于某一特定的加工顺序 π , 关键路径的指的是从第一个工件的第一道工序起一直到最后一个工件完成加工时的连续作业路径, 对应图 1 中红色路线总长度, 此长度就对应最大完工时间 C_{\max} 。(2) 反向关键路径: 由于无等待约束的存在, 个别 NWFSP 无法找到一条完整正向关键路径, 会存在如图 2 蓝色路线所示的反向关键路径时, 此时关键路径总长度 (C_{\max}) 等于正向关键路径总长度减去反向关键路径总长度^[21]。

EENWFSP 工件块定义: EENWFSP 工件块加具有如下特点: 设 $v_h > v_l$, 若工件 $j(j=1,2,\dots,n)$ 以速度 v_h 在机器 $i(i=1,2,\dots,m)$ 上加工时, 比以速度 v_l 加工时耗时更短, 但能源消耗更多。

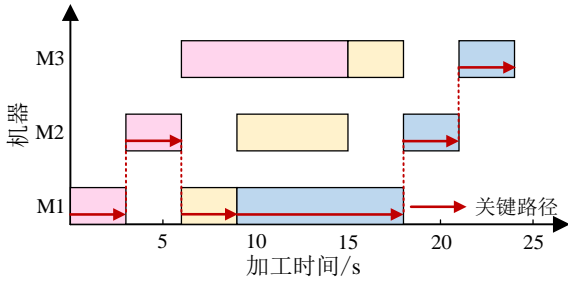


图 1 正向关键路径示意图

Fig. 1 Schematic diagram of forward critical path

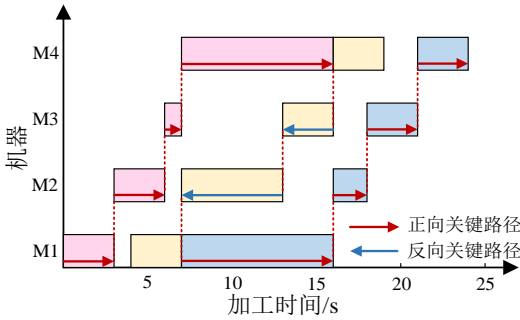


图 2 反向关键路径示意图

Fig. 2 Schematic diagram of reverse critical path

性质 1: 存在两个调度方案: $\Delta_1 = (\pi_1, V)$ 和 $\Delta_2 = (\pi_2, V)$, 满足 $C_{\max}(\Delta_1) > C_{\max}(\Delta_2)$, 则 $\Delta_1 \prec \Delta_2$ 。

证明: 已知总能耗 $TEC = TEC^s + TEC^p$, 其中 TEC^s 和 TEC^p 分别表示机器的静止能耗和加工能耗。由于两个调度方案的速度相同, 则两个调度方案的加工总能耗是相同的, 即 $TEC^p(\Delta_1) = TEC^p(\Delta_2)$ 。设 t_i^s 和 t_i^p 分别表示调度方案 Δ_k ($k=1,2$) 在机器 i 上的总静止时间和总加工时间, 则有:

$$t_i^s(\Delta_k) + t_i^p(\Delta_k) = C_{\max}(\Delta_k) \quad (16)$$

$$\forall i=1,2,\dots,m, k=1,2$$

由于 $C_{\max}(\Delta_1) > C_{\max}(\Delta_2)$ 且 $t_i^p(\Delta_1) = t_i^p(\Delta_2)$,

$\forall i=1,2,\dots,m$, 则 $t_i^s(\Delta_1) > t_i^s(\Delta_2)$, $\forall i=1,2,\dots,m$,

可得

$$TEC^s(\Delta_1) = \sum_{i=1}^m t_i^s(\Delta_1) \phi_i > \sum_{i=1}^m t_i^s(\Delta_2) \phi_i = TEC^s(\Delta_2)$$

, 因此, $TEC^s(\Delta_1) > TEC^s(\Delta_2)$, 根据已知条件

$C_{\max}(\Delta_1) > C_{\max}(\Delta_2)$, 所以 $\Delta_1 \prec \Delta_2$ 。

性质 2: 存在两个调度方案: $\Delta_1 = (\pi, V_1)$ 和 $\Delta_2 = (\pi, V_2)$, 满足 $C_{\max}(\Delta_1) = C_{\max}(\Delta_2)$, $\forall i \in \{1,2,\dots,n\}$, $j \in \{1,2,\dots,m\}$, $v_{i,j}(\Delta_1) \geq v_{i,j}(\Delta_2)$ 和 $v_{i,j}(\Delta_1) > v_{i,j}(\Delta_2)$ $j \in \{1,2,\dots,m\}$, $\exists i \in \{1,2,\dots,n\}$, 则 $TEC(\Delta_1) > TEC(\Delta_2)$ 且 $\Delta_1 \prec \Delta_2$ 。

证明: 根据式(16)且已知 Δ_1 加工速度更快, 又因为 $C_{\max}(\Delta_1) = C_{\max}(\Delta_2)$, $t_i^p(\Delta_1) \leq t_i^p(\Delta_2)$, $t_i^s(\Delta_1) \geq t_i^s(\Delta_2)$, $\forall i=1,2,\dots,m$, 可得, $TEC^s(\Delta_1) = \sum_{i=1}^m t_i^s(\Delta_1) \phi_i > \sum_{i=1}^m t_i^s(\Delta_2) \phi_i = TEC^s(\Delta_2)$, 因此 $TEC^s(\Delta_1) > TEC^s(\Delta_2)$, 由定义 2 可得 $TEC^p(\Delta_1) > TEC^p(\Delta_2)$, 则 $TEC(\Delta_1) > TEC(\Delta_2)$ 且 $\Delta_1 \prec \Delta_2$ 。

性质 3: 根据性质 2, 可给出推论: 给定调度方案 $\Delta_1 = (\pi_1, V_1)$, 在不影响完工时间的情况下适当减小非关键路径上机器的加工速度, 可降低该调度方案的总能耗。

3.5 工序优化阶段

根据性质 1 可得, 在速度固定的情况下, 对 C_{\max} 进行优化, TEC 也会随之优化, 调度方案更优。因此, 将交换、移动和对称算子用于优化工件的排序, 在这个过程中, 各工件的加工速度保持不变, 三种离散状态转移算子的操作过程如下。

3.5.1 离散状态转移算子操作

1) 交换变换算子

若 $m_a=3$, 调度方案为 $\pi=(1,2,3,4,5)^T$, 可使用如下的变换矩阵:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

使用变换矩阵之后, π 变为 $(1,5,3,2,4)^T$, 从变换后序列中可看出被交换位置的位数为 3。

2) 移动变换算子

若 $m_b=3$, 调度方案为 $\pi=(1,2,3,4,5)^T$, 可使用如下的变换矩阵:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

使用移动变换矩阵之后, π 变为 $(1,2,4,5,3)^T$, 从变换后序列中可看出被连续移动的位数为 2。

3) 对称变换算子

若 $m_c=1$, 调度方案为 $\pi=(1,2,3,4,5)^T$, 可使用如下的变换矩阵:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

使用对称变换矩阵之后, π 变为 $(1,2,5,4,3)^T$, 从变换后序列中可看出对称中心的长度为 1。

在工件排序优化阶段, 交换和移动算子主要是用于局部搜索操作, 对称算子则是用于全局搜索。

3.5.2 二次状态转移操作

为进一步提高 HDSTA 在工序优化阶段的搜索深度, 本文在一次候选解集的基础上再次进行状态转移变换, 这种操作被称作二次状态转移变换, 已经证明, 二次状态转移能够在不增加耗时的情况下, 保证候选解具有比一次状态转移更为丰富的工件序列信息, 且在一定程度上能够增加算法的稳定性和全局搜索能力^[22]。图 3 展示了工序最优解为 $[2,4,1,3]$ 时, 使用交换变换算子进行二次状态转移操作的示例, 根据该图可知, 二次状态转移可以获得信息更为丰富的候选解集。

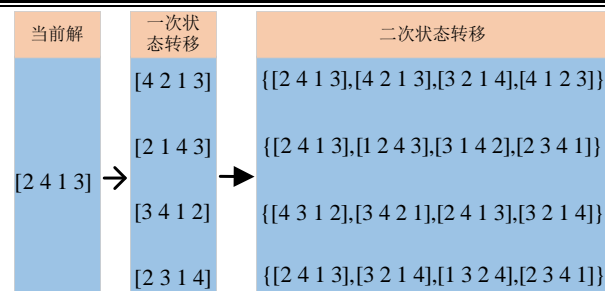


图3 二次状态转移示例

Fig. 3 Example of secondary state transition

3.6 速度替换优化阶段

通过性质 2 和性质 3 可知, 在 C_{\max} 不变的情况下, 通过适当降低工件的加工速度, 可以使 TEC 更小, 调度方案更优。

为适配 EENWFSP 调度问题特性, 本文考虑将替代变换算子用于速度优化阶段, 而速度矩阵与工件序列编码不同, 因此将替代变换算子进行调整:

$$V_{k+1} = V_k \oplus m_d \quad (17)$$

式中, $V_k \in R^{n \times m}$ 是速度矩阵, \oplus 代表替换操作; 该算子具有将 V_k 中最多 m_d 个元素进行替换的功能, 本文中 $m_d=1$ 。一般情况下, 学者会使用变异、交叉或重新分配操作对速度阵进行更新。

一般来说, 如果提高关键路径上工件的处理速度, 则 C_{\max} 会缩短, TEC 会增加。反之, 如果降低关键路径上工件的处理速度, 则会增加 C_{\max} , TEC 会减少。因此, 基于性质 2 和性质 3, 速度替换优化阶段选择适当降低非关键路径上的机器加工速度, 如图 4 所示, 随机替换一道速度矩阵中处于非关键路径上的工序, 降低其速度 (若速度为最高级别, 则可降低一档或两档, 若速度为中等速度, 则降低一档), 就可以达到在不影响完工时间的情况下降低能耗, 而进一步提升 Pareto 解的质量。为进一步提高 Pareto 解集多样性, 速度替换规模设置为 NP 。在速度优化阶段, 替换算子起到对当机器速度进行替换扰动来获得更多优质 Pareto 解的作用。

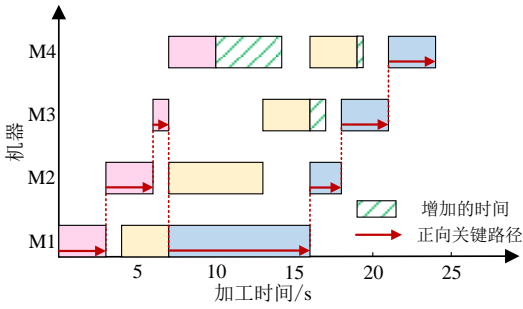


图 4 速度替换操作示意图

Fig. 4 Schematic diagram of speed replacement operation

3.7 基于破坏重构的混合优化阶段

经过上述速度替换优化阶段,此时算法种群规模为 NP ,为进一步提升 Pareto 解集质量与多样性,且结合 DSTA 也属于迭代贪婪规则的算法特点,本文嵌入了一种基于迭代贪婪算法的破坏重构的混合优化操作。破坏重构操作的破坏指的是将 x_i 中 d 个工件以及其速度向量移除,移除的工件集记作 $x_{i,R}$,剩余的工件集记作 $x_{i,D}$,给移除的工件集 $x_{i,R}$ 随机分配速度等级,然后利用重构操作将移除的工件依次插入 $x_{i,D}$ 所有可能的位置,利用非支配准则选择较优解保存。

破坏重构操作如下:

Step1: 输入当前加工序列 x_i , 破坏因子 d ;

Step2: 令 $x_{i,D} = x_i$, 从中随机移除 d 个工件,放入序列 $x_{i,R}$ 中;

Step3: 依次选择 $x_{i,R}$ 中的工件,并将各工序对应机器速度进行随机分配,将其插入 $x_{i,D}$ 中所有可能位置,以非支配准则为评判更新标准,更新序列 x_i ;

Step4: 将种群中所有个体执行 step1-step3 操作。

3.8 算法步骤

HDSTA 具体流程如下:

Step1: 设置算法参数,按照 3.2 和 3.3 节方法初始化外部档案集 AS;

Step2: 随机选择 AS 中一个候选解的工件方案作为初始解,利用 3.5 节操作进行工件排序优化;

Step3: 继续利用 Step2 中候选解的速度方案作为初始解,将种群规模扩大至 NP ,利用 3.6 节操作进行机器速度优化;

Step4: 利用 3.7 节破坏重构操作对种群进行进一步优化;

Step5: 评价更新后种群中的所有个体,更新外部档案集 AS;

Step6: 若满足算法停止条件,则输出外部档案集 AS,否则跳转至 Step2。

图 5 给出 HDSTA 流程图, gen 为迭代次数。

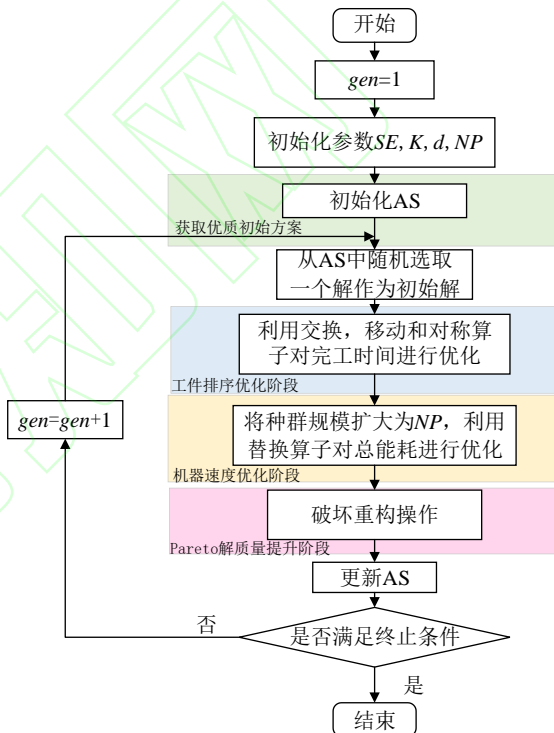


图 5 HDSTA 流程图

Fig. 5 HDSTA flow chart

4 计算实验

为验证 HDSTA 在求解 EENWFSP 调度问题的有效性,对比实验均基于 CPU 主频 2.4GHz 和 RAM 为 8GB 的计算机上使用 MATLAB 2021a 仿真平台实现。

4.1 测试实例与评价指标

本文选择经典流水车间算例 Taillard Benchmark 中各规模算例的后五个测试实例进行实验。参考文献 [23], 将速度级别设置为

$L = \{1, 2, 3\}$, 各速度级别对应的真实速度和对应单位时间内的能耗为 $v = \{1.2, 1, 0.8\}$ 和 $\tau = \{1.5, 1, 0.6\}$ 。

本文采用解集覆盖率 C , 总非支配向量数目 $ONVG$ 和指标 SP 来评价算法所得 Pareto 解集质量。

(1) 解集覆盖率 C 用于测量非支配解集 S_1 和 S_2 之间的支配关系。

$$C(S_1, S_2) = \frac{|\{u \in S_2 \mid \exists v \in S_1 : v \prec u\}|}{|S_2|} \quad (18)$$

式中, $|S_2|$ 表示 S_2 中解数量, $C(S_1, S_2)=1$ 表示 S_1 中部分解支配 S_2 中所有解, $C(S_1, S_2)=0$ 表示 S_1 中任意解都不能支配 S_2 中解, 显然, C 越大, Pareto 解集质量越好。

(2) 总非支配向量数目 $ONVG$ 用于计算解集 S 中非支配解的数量, 定义如下:

$$ONVG(NS) = |NS| \quad (19)$$

式中, NS 代表算法求解完成后解集 S 中包含的全部非支配解, $ONVG$ 越大, 算法获得的非支配解数目越多, 性能越好。

(3) 指标 SP 用于评价非支配解分布均匀程度。

$$SP(S) = \sqrt{\frac{1}{|S|} \sum_{x \in S} \frac{d_x - \bar{d}}{\bar{d}}} \quad (20)$$

式中, d_x 表示解 x 与其最近解之间的欧氏距离, $\bar{d} = \sum_{x \in S} d_x / |S|$ 表示 S 中所有解的平均距离, 显然 SP 越小, Pareto 解集分布越均匀, 质量越好。

4.2 参数设置

HDSTA 的关键参数包括替换规模 NP , 破坏重构参数 d , 外部档案集长度 K 和采样规模 SE 。为合理确定各参数的取值, 本节选择一个中等规模算例 TA60(50*20), 设计三水平的 DOE 实验, 如表 2 所示, 对于每组参数, 算法独立运行 20 次, 运行时间均为 $(n^2 / 2 \times 100)$ ms。将仿真结果合并为该组参数组合的 Pareto 解集, 再将 9 组 Pareto 解集合并, 作为 TA60 的 Pareto 解集。再将每组参数组合下 Pareto 解的数量占 TA60 的 Pareto 解集百分比作为响应变量(Response Variable, RV), 显然, RV 值越大, 参数组合越好。

表 2 参数水平设置表
Table 2 Parameter level setting table

参数	水平		
	1	2	3
NP	20	30	50
d	2	4	8
K	30	40	50
SE	30	60	100

表 3 参数正交表及其 RV 值
Table 3 Orthogonal table of parameters and its RV value

参数组	水平				$RV(\%)$
	NP	d	K	SE	
合					
1	20(1)	2(1)	30(1)	30(1)	8.25
2	20(1)	4(2)	40(2)	60(2)	7.92
3	20(1)	8(3)	50(3)	100(3)	4.95
4	30(2)	2(1)	40(2)	100(3)	15.51
5	30(2)	4(2)	50(3)	30(1)	10.23
6	30(2)	8(3)	30(1)	60(2)	10.23
7	50(3)	2(1)	50(3)	60(2)	19.14
8	50(3)	4(2)	30(1)	100(3)	15.84
9	50(3)	8(3)	40(2)	30(1)	7.92

表 4 各组参数的平均 RV
Table 4 Average RV of each group of parameters

水平	NP	d	K	SE
1	7.04	14.3	11.44	8.8
2	11.99	11.33	10.45	12.43
3	14.31	7.7	11.44	12.1
极差	7.27	6.6	0.99	3.63
等级	1	2	4	3

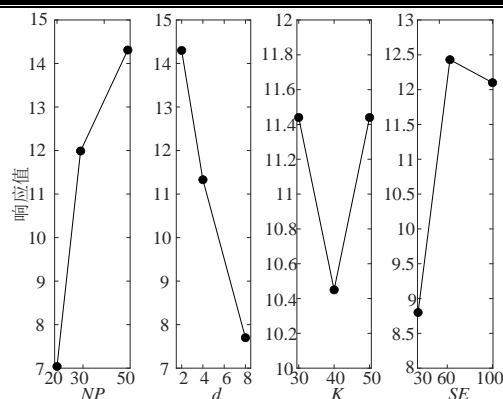


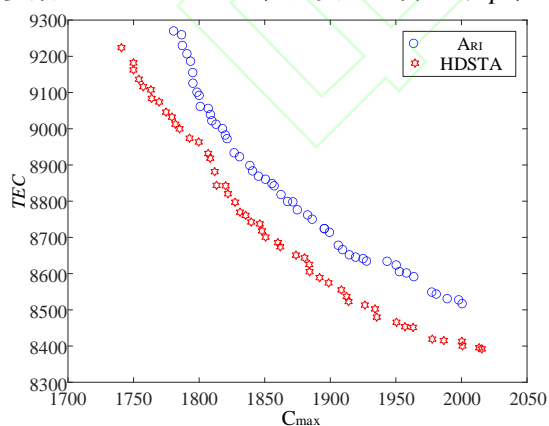
图6 参数对算法性能的影响

Fig. 6 Influence of parameters on algorithm performance

由表4和图6可以看出, HDSTA的参数设置为 $NP=50$, $d=2$, $K=50$ 和 $SE=60$ 时算法性能最好, 故选择上述参数设置用于 HDSTA 进行后续实验。

4.3 策略的有效性分析

为证明初始化方法、速度替换方法和破坏重构方法的有效性, 将 HDSTA 分别与使用随机初始化方式的 HDSTA (记作 A_{RI})、使用随机速度替换策略(随机替换指的是随机增大或者降低一级或两级机器加工的速度档位)的 HDSTA (记作 A_{RS}) 和不使用破坏重构操作的 HDSTA (记作 A_{WD}) 进行比较, 记录 Taillard 各规模算例的后五个测试实例的 C 指标(每个算例运行 10 次), 如表 5~表 7 所示。为分析各算法的差异显著性, 执行置信度为 95% 的非参数 Kruskal - Wallis 检验实验, 并记录 p 值。

图7 HDSTA 和 A_{RI} 的 Pareto 曲线(TA17)Fig. 7 Pareto curve of HDSTA and A_{RI} (TA17)表5 HDSTA 和 A_{RI} 的 C 指标对比Table 5 Comparison of C between HDSTA and A_{RI}

Instance	$C(HDSTA, A_{RI})$	$C(A_{RI}, HDSTA)$	p
20×5	0.770	0.091	0
20×10	0.761	0.010	0
20×20	0.668	0.204	0.002
50×5	0.708	0.102	0
50×10	0.467	0.118	0.005
50×20	0.828	0.248	0.001
100×5	0.460	0.106	0.006
100×10	0.261	0.086	0.018
100×20	0.608	0.148	0.004
200×10	0.629	0.071	0
200×20	0.776	0.134	0.001

根据表5中数据可以看出 $C(HDSTA, A_{RI})$ 在大多数测试实例中比 $C(A_{RI}, HDSTA)$ 大, 这表示大多数 A_{RI} 获得的非支配解都被 HDSTA 获得的非支配解支配。且所有 p 值都小于 0.05, 这表示 HDSTA 明显优于 A_{RI} 。图7给出了 HDSTA 和 A_{RI} 在求解 TA17 算例时获得的 Pareto 曲线, 可以看出 HDSTA 明显优于 A_{RI} 。因此, 本文设计的初始化方式在求解 EENWFSP 时是有效的。

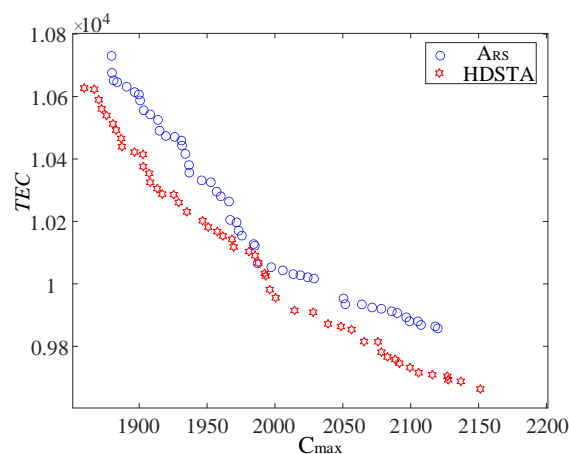
图8 HDSTA 和 A_{RS} 的 Pareto 曲线(TA20)Fig. 8 Pareto curve of HDSTA and A_{RS} (TA20)

表 6 HDSTA 和 A_{RS} 的 C 指标对比
Table 6 Comparison of C between HDSTA and A_{RS}

Instance	$C(HDSTA, A_{RS})$	$C(A_{RS}, HDSTA)$	p
20×5	0.820	0.084	0
20×10	0.864	0.052	0
20×20	0.835	0.060	0
50×5	0.936	0.176	0
50×10	0.872	0.128	0.001
50×20	0.896	0.052	0
100×5	0.704	0.056	0
100×10	0.732	0.128	0.001
100×20	0.850	0.179	0
200×10	0.772	0.072	0
200×20	0.883	0.074	0

表 7 HDSTA 和 A_{WD} 的 C 指标对比
Table 7 Comparison of C between HDSTA and A_{WD}

Instance	$C(HDSTA, A_{WD})$	$C(A_{WD}, HDSTA)$	p
20×5	0.764	0.081	0.001
20×10	0.840	0.000	0.001
20×20	0.913	0.007	0
50×5	0.852	0.062	0
50×10	0.608	0.136	0.001
50×20	0.820	0.072	0
100×5	0.850	0.064	0
100×10	0.573	0.233	0.034
100×20	0.632	0.041	0
200×10	0.750	0.171	0.001
200×20	0.816	0.113	0

根据表 6 中数据可以看出 $C(HDSTA, A_{RS})$ 在大多数测试实例中比 $C(A_{RS}, HDSTA)$ 大, 这表示大多数 A_{RS} 获得的非支配解都被 HDSTA 获得的非支配解支配。且所有 p 值都小于 0.05, 这表示 HDSTA 明显优于 A_{RS} 。图 8 给出了 HDSTA 和 A_{RS} 在求解 TA20 算例时获得的 Pareto 前沿, 可以看出 HDSTA 明显优于 A_{RS} 。因此, 本文设计的速度替换方法在求解 EENWFSP 时是有效的。

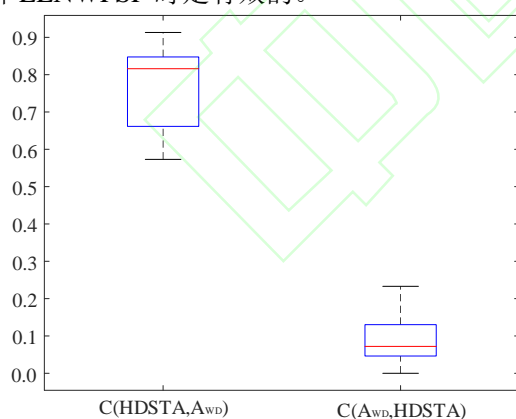


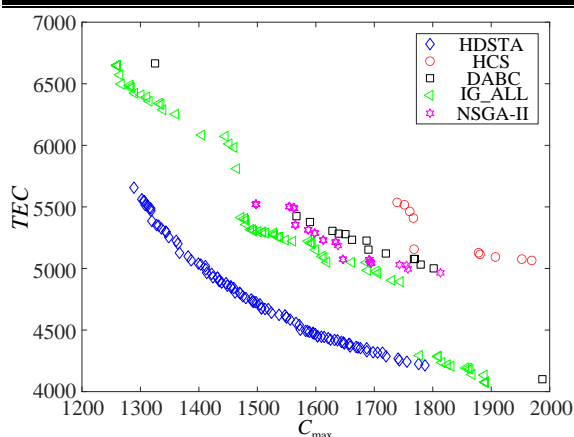
图 9 HDSTA 和 A_{WD} 的箱型图对比

Fig. 9 Box diagram of HDSTA and A_{WD}

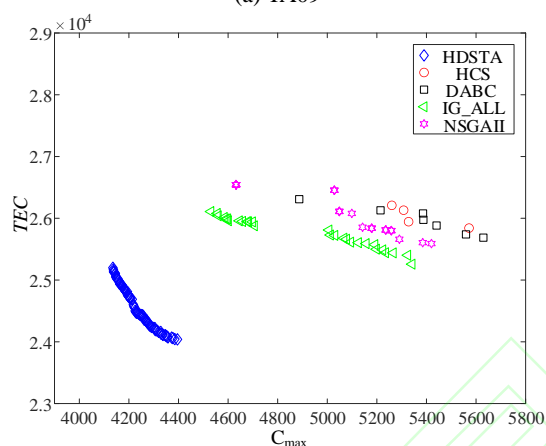
图 9 给出了 HDSTA 和 A_{WD} 求解各算例 C 指标的箱型图对比, 根据表 7 中数据和图 9 可看出 $C(HDSTA, A_{WD})$ 在大多数测试实例中比 $C(A_{WD}, HDSTA)$ 大, 这表示大多数 A_{WD} 获得的非支配解都被 HDSTA 获得的非支配解支配。且所有 p 值都小于 0.05, 这表示 HDSTA 优于 A_{WD} 。因此, 本文设计的混合破坏重构优化方式在求解 EENWFSP 时是有效的。

4.4 与其他算法性能对比分析

为验证 HDSTA 求解 EENWFSP 的有效性, 将其与经典的多目标优化算法 NSGA-II^[24], 混合布谷鸟算法(HCS)^[25], 多目标迭代贪心算法(IG_all)^[26]和离散人工蜂群算法(DABC)^[15]进行对比, 为解决 EENWFSP, 对 NSGA-II 略作改动, 在文献[24]的基础上增加速度选择操作, 速度变异操作采用随机替换操作, 与 4.3.2 节中 A_{RS} 相似。HDSTA 的参数设置如 4.2 节所示, 本文选取 55 个测试实例来测试所有算法的性能, 各算法在相同的运行环境独立运行 10 次, 运行时间均为 $(n^2 / 2 \times 100)$ ms。表 8-表 10 列出了所有规模算例的 C , $ONVG$ 和 SP 指标的平均值。



(a) TA09



(b) TA46

图 10 五种算法的 Pareto 前沿

Fig. 10 Pareto curve of five algorithms

从表 8 中可以看出, 在所有测试实例中 $C(\text{HDSTA}, \text{HCS})$ 都等于 1 且 $C(\text{HCS}, \text{HDSTA})$ 都等于 0, 这表示所有 HCS 获得的非支配解都被 HDSTA 获得的非支配解支配。相似的, 将 HDSTA 与 DABC, IG_ALL 和 NSGA-II 相比, 在所有测试实例中, $C(\text{HDSTA}, \text{DABC})$, $C(\text{HDSTA}, \text{IG_ALL})$ 和 $C(\text{HDSTA}, \text{NSGA-II})$ 都远大于 $C(\text{DABC}, \text{HDSTA})$, $C(\text{IG_ALL}, \text{HDSTA})$ 和 $C(\text{NSGA-II}, \text{HDSTA})$, 这表明 HDSTA 获得的 Pareto 解集质量优于 DABC, IG_ALL 和 NSGA-II。优势较为明显的原因是 HDSTA 相较于比较方法加入了基于问题性质的速度替换操作, 较为明显提升了算法的搜索性能。除此之外, 图 10(a)和(b)分别给出五种算法求解 TA09 和 TA46 的 Pareto 曲线, 从曲线上可以看出 HDSTA 在收敛性和分布性上优

于其余四种对比算法。

表 8 各算法的 C 指标平均值对比Table 8 Comparison of the average value of C index of each algorithm

($A_1 = C(\text{HDSTA}, \text{HCS})$ $A'_1 = C(\text{HCS}, \text{HDSTA})$; $A_2 = C(\text{HDSTA}, \text{DABC})$
 $A'_2 = C(\text{DABC}, \text{HDSTA})$; $A_3 = C(\text{HDSTA}, \text{IG_ALL})$ $A'_3 = C(\text{IG_ALL}, \text{HDSTA})$;
 $A_4 = C(\text{HDSTA}, \text{NSGA-II})$ $A'_4 = C(\text{NSGA-II}, \text{HDSTA})$)

Instance	A_1	A'_1	A_2	A'_2	A_3	A'_3	A_4	A'_4
20×5	1.00	0.00	0.98	0.00	0.87	0.00	0.99	0.00
20×10	1.00	0.00	0.83	0.00	0.76	0.00	1.00	0.00
20×20	1.00	0.00	0.63	0.00	0.77	0.00	1.00	0.00
50×5	1.00	0.00	0.71	0.00	0.56	0.00	1.00	0.00
50×10	1.00	0.00	0.71	0.00	0.48	0.00	0.96	0.00
50×20	1.00	0.00	0.63	0.00	0.49	0.00	1.00	0.00
100×5	1.00	0.00	0.58	0.00	0.53	0.00	0.95	0.00
100×10	1.00	0.00	0.67	0.00	0.45	0.00	1.00	0.00
100×20	1.00	0.00	0.65	0.00	0.46	0.00	1.00	0.00
200×10	0.96	0.00	0.55	0.00	0.84	0.00	1.00	0.00
200×20	0.98	0.00	0.71	0.00	0.72	0.00	1.00	0.00
Average	0.99	0.00	0.70	0.00	0.63	0.00	0.99	0.00

从表 9 中数据和图 11 中各算法的平均 ONVG 箱形图可以看出, HDSTA 的 ONVG 指标远大于 HCS, DABC 和 NSGA-II, 略大于 IG_ALL。在工件数目为 20 的小规模算例中, HDSTA 除了在工件数目为 10 时的 ONVG 指标略差于 IG_ALL, 整体上优于其余三种对比方法; 且随着算例规模的增大, 虽然在个别中等规模算例略差于 IG_ALL, 但是差距不大, 但在大规模算例中, HDSTA 是表现较为突出的, 这表明 HDSTA 相较于对比算法具有较好的稳定性, 能够获得多样的 Pareto 解集。

除此之外, 从表 10 中数据以及图 12 中各算法的平均 SP 箱形图可以看出, HDSTA 的 SP 指标最优, 可以得出 HDSTA 所获得的 Pareto 解集的分布性优于其余四种算法。四种算法中 HCS 的 Pareto 解集的收敛性和分布性相对欠佳, 且 HCS 和 DABC 解的多样性相对较差。由以上分析可以得出 HDSTA 在求解 EENWFSP 调度问题时优于其余四种算法。因此, HDSTA 是求解 EENWFSP 具有较

强竞争力的方法。

表9 各算法的 *ONVG* 平均值对比
Table 9 *ONVG* average comparison of five algorithm

Instance	HDSTA	HCS	DABC	IG_ALL	NSGA-II
20×5	85.20	12.80	18.80	66.60	43.20
20×10	73.80	13.00	34.60	83.60	22.40
20×20	60.00	12.20	35.00	48.60	25.40
50×5	73.00	28.25	36.28	84.25	35.20
50×10	74.60	17.00	25.61	58.02	34.00
50×20	41.00	8.60	15.10	42.20	11.60
100×5	66.40	15.10	17.24	84.80	43.40
100×10	68.70	19.20	9.21	58.20	23.32
100×20	49.80	12.20	12.15	42.20	18.60
200×10	36.20	8.00	37.84	27.00	20.00
200×20	47.60	9.20	18.54	36.80	11.71
Average	61.48	14.14	23.67	57.48	26.26

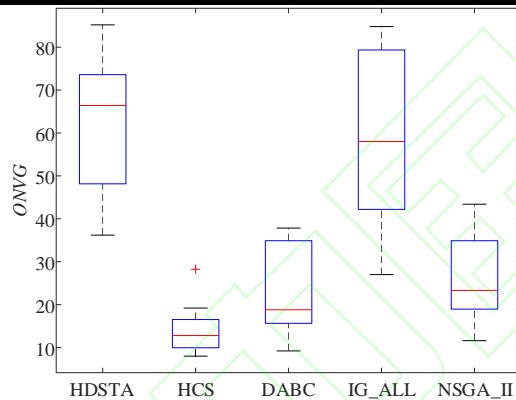


图11 五种算法的 *ONVG* 箱型图对比

Fig. 11 Comparison of *ONVG* box diagram of five algorithms

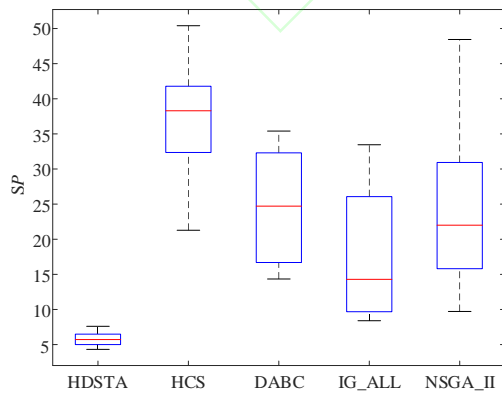


图12 五种算法的 *SP* 箱型图对比

Fig. 12 Comparison of *SP* box diagram of five algorithms

表10 各算法的 *SP* 平均值对比
Table 10 *SP* average comparison of five algorithm

Instance	HDSTA	HCS	DABC	IG_ALL	NSGA-II
20×5	5.99	23.40	15.10	8.40	9.72
20×10	5.72	33.6	24.71	14.29	14.91
20×20	4.99	34.37	32.57	26.96	18.49
50×5	7.04	21.28	31.24	9.31	19.27
50×10	6.07	49.57	32.53	9.78	25.66
50×20	5.12	40.61	17.84	21.13	32.69
100×5	7.60	50.40	31.60	13.58	22.00
100×10	4.32	39.12	14.34	9.65	23.20
100×20	5.05	31.94	24.31	23.40	41.49
200×10	6.63	38.29	16.30	29.62	48.44
200×20	4.93	42.18	35.40	33.45	11.38
Average	5.77	36.80	25.09	18.14	24.30

5 结论

本文基于 *DSTA* 提出了一种混合离散状态转移算法(*HDSTA*)求解 *EENWFSP* 调度问题。*NEH* 改进方法获得优质的初始调度方案, 根据 *EENWFSP* 特性合理分配四个离散操作算子的优化环节, 较好地平衡了算法的全局搜索和局部搜索能力, 将交换、移动和对称算子用于工件排序优化, 替换算子用于速度优化。增加基于关键路径的替换操作算子以及最后加入破坏重构操作混合优化环节, 进一步提高 *Pareto* 解集质量和多样性。仿真结果表明, *HDSTA* 各部分的改进方式效果明显, 且 *HDSTA* 相较于其余四种算法在求解 *EENWFSP* 时在 3 种指标上的性能表现, 可以看出 *HDSTA* 获得的非支配解数目更多, 收敛性更好且分布也更均匀。未来将进一步研究 *DSTA* 在分布式无等待低碳调度问题以及不确定低碳调度问题等复杂调度问题的应用。

参考文献

- [1] Gahm C, Denz F, Dirr M, et al. Energy-efficient scheduling in manufacturing companies: A review and research framework[J]. *European Journal of Operational Research*(S0377-2217), 2016, 248(03): 744–757.
- [2] 王凌, 王晶晶, 吴楚格. 绿色车间调度优化研究进展[J]. *控制与决策*, 2018, 33(03): 385–391.
Wang Ling, Wang Jingjing, Wu Chuge. Advances in green shop scheduling and optimization[J]. *Control and Decision*, 2018, 33(03): 385–391.
- [3] Lei D M, Gao L, Zheng Y. A novel teaching-learning based optimization algorithm for energy-efficient scheduling in hybrid flow shop[J]. *IEEE Transactions on Engineering Management*(S 1558-0040), 2018, 65(02): 330–340.
- [4] Gu W B, Li Z, Dai M, et al. An energy-efficient multi-objective permutation flow shop scheduling problem using an improved hybrid cuckoo search algorithm[J]. *Advances in Mechanical Engineering*(S1687-8132). 2021, 13(06): 1–15.
- [5] 任彩乐, 杨旭东, 张超勇, 等. 面向节能的混合流水车间调度问题建模与优化[J]. *计算机集成制造系统*, 2019, 25(08): 1965–1980.
Ren Caile, Yang Xudong, Zhang Chaoyong, et al. Modeling and optimization for energy-efficient hybrid flow shop scheduling problem[J]. *Computer Integrated Manufacturing Systems*, 2019, 25(08): 1965–1980.
- [6] 艾子义, 雷德明. 基于新型蛙跳算法的低碳柔性作业车间调度[J]. *控制理论与应用*, 2017, 34(10): 1361–1368.
Ai Ziyi, Lei Deming. A novel shuffled frog leaping algorithm for low carbon flexible job shop scheduling[J]. *Control Theory & Applications*, 2017, 34(10): 1361–1368.
- [7] Tang D B, Dai M, Miguel A, et al. Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization[J]. *Computers in Industry*(S0166-3615), 2016, 81(C): 82–95.
- [8] Shao W, Pi D, Shao Z. An extended teaching-learning based optimization algorithm for solving no-wait flow shop scheduling problem[J]. *Applied Soft Computing*(S1568-4946), 2017, 61: 193–210.
- [9] Zhao F, Liu H, Zhang Y, et al. A discrete water wave optimization algorithm for no-wait flow shop scheduling problem[J]. *Expert Systems with Applications*(S0957-4174), 2018, 91: 347–363.
- [10] Wei S H, Lin, Ching K, et al. Optimization of makespan for no-wait flowshop scheduling problems using efficient matheuristics[J]. *Omega*(S0030-2228), 2016, 64: 115–125.
- [11] Benini L, Bogliolo A, De Micheli G. A survey of design techniques for system-level dynamic power management[J]. *IEEE Transactions on Very Large Scale Integration Systems*(S1063-8210), 2000, 8(03): 299–316.
- [12] Liu C H, Huang D H. Reduction of power consumption and carbon footprints by applying multi-objective optimisation via genetic algorithms[J]. *International Journal of Production Research*(S0020-7543), 2014, 52(02): 337–352.
- [13] Ding J Y, Song S J, Wu C, et al. Carbon-efficient scheduling of flow shops by multi-objective optimization[J]. *European Journal of Operational Research*(S0377-2217), 2016, 248(03): 758–771.
- [14] Jiang E D, Wang L. An improved multi-objective evolutionary algorithm based on decomposition for energy-efficient permutation flow shop scheduling problem with sequence-dependent setup time[J]. *International Journal of Production Research*(S0020-7543), 2019, 57(06): 1756–1771.
- [15] Tasgetiren M F, D Y üksel, Gao L, et al. A discrete artificial bee colony algorithm for the energy-efficient no-wait flowshop scheduling problem[J]. *Procedia Manufacturing*(S2351-9789), 2019, 39: 1223–1231.
- [16] Wang J J, Wang L. A knowledge-based cooperative algorithm for energy-efficient scheduling of distributed flow-shop[J]. *IEEE Transactions on Systems*(S0018-9472). 2020, 50(05): 1805–1819.
- [17] 潘子肖, 雷德明. 基于问题性质的分布式低碳并行机调度算法研究[J]. *自动化学报*, 2020, 46(11): 2427–2438.
Pan Zixiao, Lei Deming. Research on property-based distributed low carbon parallel machines scheduling algorithm[J]. *Acta Automatica Sinica*, 2020, 46(11): 2427–2438.
- [18] Zhou X J, Yang C H, Gui W H. State transition algorithm[J]. *Journal of Industrial and Management Optimization*(S1547-5816), 2012, 8(04): 1039–1056.
- [19] 阳春华, 唐小林, 周晓君, 等. 一种求解旅行商问题的离散状态转移算法[J]. *控制理论与应用*, 2013, 30(08): 1040–1046.
Yang Chunhua, Tang Xiaolin, Zhou Xiaojun, et al. A discrete state transition algorithm for traveling salesman problem[J]. *Control Theory & Applications*, 2013, 30(08): 1040–1046.
- [20] Zhou X J, Gao D, Simpson A. Optimal design of water distribution networks by a discrete state transition algorithm[J]. *Engineering Optimization*(S0305-215X), 2015, 48(04): 1–26.
- [21] Gui L, Gao L, Li X Y. Anomalies in special

- permutation flow shop scheduling problems[J]. Chinese Journal Of Mechanical Engineering(S1000-9345), 2020, 33(03):46-46.
- [22] 董天雪, 阳春华, 周晓君, 等. 一种求解企业员工指派问题的离散状态转移算法[J]. 控制理论与应用, 2016, 33(10): 1378-1388.
- Dong Tianxue, Yang Chunhua, Zhou Xiaojun, et al. A novel discrete state transition algorithm for staff assignment problem[J]. Control Thetory and Applications, 2016, 33(10): 1378-1388.
- [23] Mansouri S A, Aktas E, Besikci U. Green scheduling of a two-machine flowshop:trade-off between makespan and energy consumption[J]. European Journal of Operational Research(S0377-2217), 2016, 248(03): 772-788.
- [24] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation(S1089-778X), 2002, 6(02):182-197.
- [25] 钟陵充, 钱斌, 胡蓉, 等. 混合布谷鸟算法求解绿色流水车间调度问题[J]. 中国机械工程, 2018, 29(22):2674-2681.
- Zhong Lingchong, Qian Bin, Hu Rong, et al. HCS algorithm for multi-objective flow shop scheduling problem with energy consumption[J]. China Mechanical Engineering, 2018, 29(22):2674-2681.
- [26] Ztop H, Tasgetiren M F, Deniz T E, et al. Green permutation flow shop scheduling: A trade- off- between energy consumption and total flow time[C]// Proceedings of the 14th International Conference, ICIC 2018, Wuhan, China. 2018: 753-759.