

引用:刘超,韩玉艳,杨晓宇,等. 多目标协同进化算法求解具有可变加工速度和自动导引车的柔性作业车间调度问题[J]. 聊城大学学报(自然科学版),2025,38(3):317-332.

Citation:LIU Chao,HAN Yuyan,YANG Xiaoyu,et al. A multi-objective cooperative coevolutionary algorithm for solving the flexible job shop scheduling problem with variable processing speeds and automated guided vehicles[J]. Journal of Liaocheng University (Natural Science Edition),2025,38(3):317-332.

文章编号 1672-6634(2025)03-0317-16

DOI 10.19728/j.issn1672-6634.2024050009

多目标协同进化算法求解具有可变加工速度和自动导引车的柔性作业车间调度问题

刘超,韩玉艳,杨晓宇,李朕,王亚楠,王旭

(聊城大学 计算机学院,山东 聊城 252059)

摘要 针对具有自动导引车和可变加工速度约束的多目标柔性车间调度问题(Flexible Job Shop Scheduling Problem with Automated Guided Vehicles, FJSP-AGVs),首先基于位置的建模方法,设计了以最大完工时间和总能耗为优化目标的混合整数线性规划模型(MILP);然后根据问题特性,将 FJSP-AGVs 问题分解为工序排序,机器分配,加工速度设置和自动导引车分配 4 个子问题,并采用多目标协同进化算法(Multi-Objective Cooperative Coevolutionary Algorithm, MOCEA)进行求解。在 MOCEA 算法中,通过两种基于关键路径的协同进化策略促进不同种群之间的信息交互,并采用降低非关键路径上工序的加工速度策略进一步优化总能耗指标。最后,通过实验验证了所提 MOCEA 的有效性。

关键词 柔性作业车间调度;自动导引车;可变加工速度;多目标;协同进化

中图分类号 TP18;TH165

文献标志码 A

开放科学(资源服务)标识码(OSID)



A multi-objective cooperative coevolutionary algorithm for solving the flexible job shop scheduling problem with variable processing speeds and automated guided vehicles

LIU Chao,HAN Yuyan,YANG Xiaoyu, LI Zhen,WANG Yanan,WANG Xu

(School of Computer Science, Liaocheng University, Liaocheng 252059, China)

Abstract For the multi-objective Flexible Job Shop Scheduling Problem with Automated Guided Vehicles (FJSP-AGVs) and variable processing speed constraints, a mixed integer linear programming (MILP) model is first designed by using a position-based modeling approach, with the objectives of minimizing makespan and total energy consumption (TEC). Then decomposed the FJSP-AGVs problem into four interrelated subproblems according to its characteristics: operation sequencing, machine allocation, process-

收稿日期:2024-05-30

基金项目:国家自然科学基金项目(61973203,62106073);山东省自然科学基金项目(ZR2023MF022);聊城大学光岳青年创新团队项目(LCUGYTD2022-03)资助

通信作者:韩玉艳,女,汉族,博士,副教授,研究方向:智能优化调度,E-mail:hanyuyan@lcu-cs.com。

ing speed settings, and automated guided vehicle allocation. To address these four subproblems, we proposed a multi-objective cooperative coevolutionary algorithm (MOCEA). In the MOCEA, two collaborative evolution strategies based on the critical path are used to promote information exchange between different populations, and a strategy to reduce the processing speed of operations on non-critical paths is employed to further optimize the total energy consumption. Finally, experiments were conducted to validate the effectiveness of the proposed MOCEA.

Keywords flexible job shop scheduling; automated guided vehicles; variable processing speeds; multi-objective; cooperative coevolutionary

0 引言

在全球环境问题日益严峻的背景下,绿色制造备受关注^[1]。它通过减少资源和能源消耗来减轻环境负担,并推动制造业可持续发展^[2]。合理的车间调度可以提高生产效率和降低成本^[3],其中改变机器加工速度可以灵活控制生产过程,优化能耗和生产周期。因此,研究如何调整机器加工速度以优化能耗和生产周期成为绿色制造中的重要课题^[4]。

柔性作业车间调度问题(Flexible Job Shop Scheduling Problem, FJSP)广泛存在于半导体制造,汽车装配和纺织生产等领域^[5]。目前,已有文献针对 FJSP 考虑了不同约束,如离散工序序列约束^[6]、分布式操作约束^[7]、分时定价和计划停机条件^[8]、可变加工速度约束^[1]以及机器故障^[9]等。然而,在实际的柔性作业车间生产中,除了上述约束外,需要考虑跨机器传送工件的问题,即运输资源约束。自动导引车(Automated Guided Vehicles, AGVs)的出现很好地解决了这一运输资源问题,作为智能物流运输设备,AGVs 正为现代制造业的发展和生产提供重要支撑和保障^[10]。将 AGVs 的运输任务整合到 FJSP 中,形成了 FJSP-AGVs 问题,该调度模式可提高生产效率和资源利用率。

针对 FJSP-AGVs 问题的不同优化目标,如最小化完工时间^[11-12]、能耗^[13]、碳足迹和机器负载^[14],学者们提出了诸多求解方法,如深度强化学习^[15]、双种群协作遗传算法^[12]、混合遗传算法结合局部邻域搜索^[13]、以及 NSGA-II 算法^[14]等。尽管已有大量研究关注 FJSP-AGVs 问题的调度问题,但从实际生产需求出发,研究以能耗为指标的可变加工速度 FJSP-AGVs 问题相对较少,有必要构建该问题的混合整数线性规划模型和设计相应的智能优化方法。

针对具有可变加工速度约束的 FJSP-AGVs 问题,需要处理四个相互关联的子问题:工序顺序的安排,机器的分配,加工机器速度的设置和 AGVs 的分配。考虑到这些子问题之间的紧密关系,本文选择了合作型协同进化算法(Cooperative Coevolutionary Algorithm, CCEA)。该算法核心思想是通过分解问题,信息共享和设计合作策略等方式,有效解决可分解的复杂优化问题。该算法已被成功应用于求解流水车间调度^[16-17]和序列相关的调度^[18-20]。然而,目前尚未将该方法应用于解决多目标 FJSP-AGVs 问题。

本文的主要贡献如下:

- (1) 利用基于位置的建模方法,构建了多目标 FJSP-AGVs 问题的混合整数线性规划(Mixed Integer Linear Programming, MILP)模型;
- (2) 提出了两种基于关键路径的序列扰动策略,以提高算法探索解空间的能力;
- (3) 提出了一种基于非关键路径的邻域扰动策略,有效降低了能源消耗;
- (4) 为了充分评估所提出的 MOCEA 的性能,本文在经典的基准实例上进行了仿真实验,实验的结果和分析表明,所提出的 MOCEA 能够有效地解决多目标 FJSP-AGVs 问题。

1 具有可变加工速度和自动导引车约束的柔性作业车间调度数学模型

具有可变加工速度约束的 FJSP-AGVs 问题描述如下:假设有 n 个工件, δ 台机器, ζ 辆 AGV,每个工件都需要 AGV 运送到不同机器上加工,每个工件至少有一道工序,每道工序有一个或多个可用的加工机器。工序的实际加工时间受机器加工速度的影响,每台机器都有 θ 个加工速度。问题遵循以下条件:(1) 初始时

刻,所有工件和 AGV 位于装卸区 (loading/unloading area, LU), 且 AGV 和机器处于空闲状态。(2) 每道工序只能在一台机器加工, 且每台机器一次仅能加工一道工序。(3) 每台机器具备无限大的缓冲区, 存储已完工的工序。(4) AGV 速度恒定, 且运输时间与机器间的距离相关。(5) 运输完成后, AGV 可停靠在当前机器旁等待下一个运输任务。(6) 相邻工序在同一机器上加工, 不需要 AGV 运输。(7) 在运输过程中, 不考虑 AGV 的故障, 碰撞及延迟。表 1 和表 2 分别给出了相应符号、决策变量及其解释。

表 1 符号及相应解释

Tab. 1 Symbols and Corresponding Explanations

Symbols	Explanations	Symbols	Explanations
n	工件总数	v	AGV 编号
I	工件集合, $I = \{1, 2, \dots, n\}$	L	AGV 的可能运输的趟数集合, $L = \{1, 2, \dots, \sum_{i=1}^n n_i\}$
i, i'	工件编号	l	运输的趟数编号
n_i	工件 i 的工序数	$t_{m,m'}$	机器 m 到机器 m' 的运输时间
$O_{i,j}$	工件 i 的第 j 个工序	$t_{0,m}$	LU 区域到机器 m 的运输时间
j, j'	工序编号	$t_{m,0}$	机器 m 到 LU 区域的运输时间
J_i	工件 i 的工序集合, $J_i = \{1, 2, \dots, n_i\}$	h	一个足够大的正数
δ	机器总数	θ	加工速度水平总数
M	机器集合, $M = \{1, 2, \dots, \delta\}$	s	加工速度水平编号
m	机器编号	S	加工速度水平集合, $S = \{1, 2, \dots, \theta\}$
$M_{i,j}$	工序 $O_{i,j}$ 的加工机器集合, $M_{i,j} \subseteq M$	R_s	在速度水平 s 下机器的实际加工速度, $s \in S$
$p_{i,j,m}$	工序 $O_{i,j}$ 在机器 m 上的标准加工时间	α_s	在速度水平 s 下机器的单位时间加工能耗
ζ	AGV 总数	ie	单位时间空闲能耗
V	AGV 集合, $V = \{1, 2, \dots, \zeta\}$	te	单位时间运输能耗

表 2 相关决策变量及解释

Tab. 2 Decision Variables and Corresponding Explanations

Decision Variables	Explanations	Decision Variables	Explanations
$x_{i,j,m}$	二进制决策变量, 如果 $O_{i,j}$ 安排在机器 m 上则为 1, 否则为 0	$z_{i,j,m,s}$	二进制决策变量, 如果 $O_{i,j}$ 安排在速度水平为 s 的机器 m 上, 则为 1, 否则为 0
$y_{i,j,i',j',m}$	二进制决策变量, 如果 $O_{i,j}$ 和 $O_{i',j'}$ 都安排在机器 m 上加工, 并且 $O_{i,j}$ 在 $O_{i',j'}$ 之前进行加工, 则为 1, 否则为 0	$w_{i,j,v,l}$	二进制决策变量, 如果 $O_{i,j}$ 被 AGV v 的第 l 次行程运输, 则为 1, 否则为 0
$c_{i,j}$	工序 $O_{i,j}$ 的完工时间	$TE_{i,j}$	工序 $O_{i,j}$ 的运输能耗
$st_{i,j}$	工序 $O_{i,j}$ 的开始加工时间	IE_m	机器 m 的总空闲能耗
co_m	机器 m 上最后一道工序的完成时间	$PE_{i,j}$	工序 $O_{i,j}$ 的加工能耗
$a_{i,j}$	工序 $O_{i,j}$ 到达其加工机器的时间	TEC	总能耗
$b_{i,j}$	工序 $O_{i,j}$ 的运输时间	$C_{\max}$	调度的完工时间
$T_{i,j,m,s}$	工序 $O_{i,j}$ 在速度水平为 s 的机器 m 上的实际加工时间		

目标

$$\min(C_{\max}, TEC); \quad (1)$$

约束

$$\sum_{m \in M_{i,j}} x_{i,j,m} = 1, \forall i \in I, \forall j \in J_i; \quad (2)$$

$$\sum_{s \in S} z_{i,j,m,s} = x_{i,j,m}, \forall i \in I, \forall j \in J_i, \forall m \in M_{i,j}; \quad (3)$$

$$y_{i,j,i',j',m} + y_{i',j',i,j,m} \leq x_{i,j,m}, \forall i \in I, \forall i' \in I, \forall j \in J_i, \forall j' \in J_{i'}, i < i' \vee (i = i' \wedge j < j'), \quad (4)$$

$$\forall m \in M_{i,j} \cap M_{i',j'};$$

$$y_{i,j,i',j',m} + y_{i',j',i,j,m} \leq x_{i',j',m}, \forall i \in I, \forall i' \in I, \forall j \in J_i, \forall j' \in J_{i'}, i < i' \vee (i = i' \wedge j < j'), \quad (5)$$

$$\forall m \in M_{i,j} \cap M_{i',j'};$$

$$y_{i,j,i',j',m} + y_{i',j',i,j,m} \geq x_{i,j,m} + x_{i',j',m} - 1, \forall i \in I, \forall i' \in I, \forall j \in J_i, \forall j' \in J_{i'}, \quad (6)$$

$$i < i' \vee (i = i' \wedge j < j'), \forall m \in M_{i,j} \cap M_{i',j'};$$

$$T_{i,j,m,s} = (p_{i,j,m} \cdot z_{i,j,m,s}) / R_s, \forall i \in I, \forall j \in J_i, \forall m \in M_{i,j}, \forall s \in S; \quad (7)$$

$$c_{i,j} \geq c_{i,j-1} + \sum_{s \in S} \sum_{m \in M_{i,j}} (z_{i,j,m,s} \cdot T_{i,j,m,s}), \forall i \in I, \forall j \in J_i - \{1\}; \quad (8)$$

$$c_{i',j'} \geq c_{i,j} + \sum_{s \in S} (z_{i',j',m,s} \cdot T_{i',j',m,s}) + (y_{i,j,i',j',m} - 1) \cdot h, \forall i \in I, \forall i' \in I, \forall j \in J_i, \forall j' \in J_{i'}, \quad (9)$$

$$i \neq i' \vee (i = i' \wedge j \neq j'), \forall m \in M_{i,j} \cap M_{i',j'};$$

$$\sum_{v \in V} \sum_{l \in L} w_{i,1,v,l} = 1, \forall i \in I; \quad (10)$$

$$\sum_{v \in V} \sum_{l \in L} w_{i,j,v,l} = 1 - \sum_{m \in M_{i,j-1} \cap M_{i,j}} y_{i,j-1,i,j,m}, \forall i \in I, \forall j \in J_i - \{1\}; \quad (11)$$

$$\sum_{i \in I} \sum_{j \in J_i} w_{i,j,v,l} \leq 1, \forall v \in V, \forall l \in L; \quad (12)$$

$$\sum_{i \in I} \sum_{j \in J_i} w_{i,j,v,l-1} \geq \sum_{i \in I} \sum_{j \in J_i} w_{i,j,v,l}, \forall v \in V, \forall l \in L - \{1\}; \quad (13)$$

$$\sum_{l' \in L, l' < l} w_{i,j',v,l'} \leq 1 - w_{i,j,v,l}, \forall i \in I, \forall j \in J_i, \forall j' \in J_i, j' > j, \forall v \in V, \forall l \in L - \{1\}; \quad (14)$$

$$c_{i,j} \geq a_{i,j} + \sum_{m \in M_{i,j}} \sum_{s \in S} (z_{i,j,m,s} \cdot T_{i,j,m,s}), \forall i \in I, \forall j \in J_i; \quad (15)$$

$$a_{i,j} \geq c_{i,j-1} + t_{m',m} + (x_{i,j,m} + x_{i,j-1,m'} + \sum_{v \in V} \sum_{l \in L} w_{i,j,v,l} - 3) \cdot h, \forall i \in I, \forall j \in J_i - \{1\}, \quad (16)$$

$$\forall m \in M_{i,j}, \forall m' \in M_{i,j-1}, m \neq m';$$

$$a_{i',j'} \geq a_{i,j} + t_{m'',m'} + (x_{i,j,m} + x_{i',j',m'} + x_{i',j'-1,m''} + w_{i,j,v,l-1} + w_{i',j',v,l} - 5) \cdot h, \quad (17)$$

$$\forall i \in I, \forall i' \in I, \forall j \in J_i, \forall j' \in J_{i'} - \{1\}, i \neq i' \vee (i = i' \wedge j < j'), \forall m \in M_{i,j},$$

$$\forall m' \in M_{i',j'}, \forall m'' \in M_{i',j'-1}, m' \neq m'', \forall v \in V, \forall l \in L - \{1\};$$

$$a_{i',1} \geq a_{i,j} + t_{m,0} + t_{0,m'} + (x_{i,j,m} + x_{i',1,m'} + w_{i,j,v,l-1} + w_{i',1,v,l} - 4) \cdot h, \forall i \in I, \forall i' \in I, \quad (18)$$

$$i \neq i', \forall j \in J, \forall m \in M_{i,j}, \forall m' \in M_{i',1}, \forall v \in V, \forall l \in L - \{1\};$$

$$a_{i,j} \geq t_{0,m'} + t_{m',m} + (x_{i,j,m} + x_{i,j-1,m'} + \sum_{v \in V} w_{i,j,v,1} - 3) \cdot h, \forall i' \in I, \forall j' \in J_{i'} - \{1\}, \quad (19)$$

$$\forall m \in M_{i,j}, \forall m' \in M_{i,j-1}, m \neq m';$$

$$a_{i,1} \geq t_{0,m} + (x_{i,1,m} + \sum_{v \in V} w_{i,1,v,1} - 2) \cdot h, \forall i \in I, \forall m \in M_{i,1}; \quad (20)$$

$$b_{i',j'} \geq t_{m'',m'} + (x_{i,j,m} + x_{i',j',m'} + x_{i',j'-1,m''} + w_{i,j,v,l-1} + w_{i',j',v,l} - 5) \cdot h,$$

$$\forall i \in I, \forall i' \in I, \forall j \in J_i, \forall j' \in J_{i'} - \{1\}, i \neq i' \vee (i = i' \wedge j < j'), \forall m \in M_{i,j}, \quad (21)$$

$$\forall m' \in M_{i',j'}, \forall m'' \in M_{i',j'-1}, m' \neq m'', \forall v \in V, \forall l \in L - \{1\};$$

$$b_{i',1} \geq t_{m,0} + t_{0,m'} + (x_{i,j,m} + x_{i',1,m'} + w_{i,j,v,l-1} + w_{i',1,v,l} - 4) \cdot h, \forall i \in I, \forall i' \in I, \quad (22)$$

$$i \neq i', \forall j \in J, \forall m \in M_{i,j}, \forall m' \in M_{i',1}, \forall v \in V, \forall l \in L - \{1\};$$

$$b_{i,j} \geq t_{0,m'} + t_{m',m} + (x_{i,j,m} + x_{i,j-1,m'} + \sum_{v \in V} w_{i,j,v,1} - 3) \cdot h, \forall i' \in I, \forall j' \in J_{i'} - \{1\}, \quad (23)$$

$$\forall m \in M_{i,j}, \forall m' \in M_{i,j-1}, m \neq m';$$

$$b_{i,1} \geq t_{0,m} + (x_{i,1,m} + \sum_{v \in V} w_{i,1,v,1} - 2) \cdot h, \forall i \in I, \forall m \in M_{i,1}; \quad (24)$$

$$co_m \geq c_{i,j} + (x_{i,j,m} - 1) \cdot h, \forall i \in I, \forall j \in J_i, \forall m \in M_{i,j}; \quad (25)$$

$$st_{i,j} = c_{i,j} - \sum_{m \in M_{i,j}} \sum_{s \in S} (z_{i,j,m,s} \cdot T_{i,j,m,s}), \forall i \in I, \forall j \in J_i; \quad (26)$$

$$C_{\max} \geq c_{i,n_i}, \forall i \in I; \quad (27)$$

$$PE_{i,j} = \sum_{m \in M_{i,j}} (\sum_{s \in S} (z_{i,j,m,s} \cdot \alpha_s) \cdot x_{i,j,m} \cdot p_{i,j,m}), \forall i \in I, \forall j \in J_i; \quad (28)$$

$$TE_{i,j} = b_{i,j} \cdot te, \forall i \in I, \forall j \in J_i; \quad (29)$$

$$IE_m \geq co_m - \sum_{i \in I} \sum_{j \in J_i} \sum_{s \in S} (z_{i,j,m,s} \cdot T_{i,j,m,s} \cdot x_{i,j,m}) \cdot ie, \forall m \in \delta; \quad (30)$$

$$TEC = \sum_{i \in I} \sum_{j \in J_i} PE_{i,j} + \sum_{i \in I} \sum_{j \in J_i} TE_{i,j} + \sum_{m \in \delta} IE_m, \forall i \in I, \forall j \in J_i. \quad (31)$$

公式(1)给出了本文最小化的两个目标总能耗和完成时间。约束中,公式(2)表示每道工序只能有一个加工速度水平。公式(3)~(5)表示同一机器上的工序需遵循前后顺序。公式(6)表示实际加工时间受约束。公式(7)表示相邻工序的完成时间需满足特定条件。公式(8)表示不能出现加工重叠。公式(9)表示第一道工序由 AGV 运输。公式(10)表示同一机器上的相邻工序之间不需要 AGVs 运输。公式(11)~(14)表示 AGV 运输规则。公式(15)~(17)表示完成时间和到达时间的约束。公式(18)~(21)表示车辆资源限制。公式(22)~(25)表示工序的运输时间确定。公式(26)、(27)表示确定各机器上最后一道工序的完成时间和每道工序的开始加工时间。公式(28)表示满足最大完成时间的约束。公式(29)~(31)表示计算各种能耗。

2 协同进化算法

2.1 编码及解码

多目标 FJSP-AGVs 问题包括工序排序,机器分配,机器加工速度分配和 AGV 分配四个子问题。本文采用三层编码方式解决前三个子问题。即,第一层为工序序列向量(Operation sequence, OS),用于解决工序排序问题;第二层为机器序列向量(Machine sequence, MS),用于处理机器分配问题;第三层为速度序列向量(Speed sequence, SS),用于设置机器的加工速度。编码结构如图 1 所示。在 OS 中,第五个值为 2,且该值从左到右时第 1 次出现,因此该工序为 $O_{2,1}$ 。在 MS 中,第四个值为 1,表示工序 $O_{2,1}$ 在机器 1 加工。在 SS 中,最后三个值分别为 2,3,4,对应于 $O_{3,1}$, $O_{3,2}$ 和 $O_{3,3}$ 所选的机器加工速度。AGV 的分配,Makespan(完工时间)和 TEC(总能耗)值的计算均在解码中完成。首先,对于每道工序,选择最早到达该工序的 AGV。然后,更新 AGV 的可用时间,当前位置及运输能耗。接下来,计算每道工序的完成时间和能耗。最后,更新机器的可用时间。算法 1 给出了详细的解码过程。其中,10~26 行为 AGV 的分配过程和工件运输能耗的计算。第 26~48 行更新工序的完成时间,机器的可用时间,工序的加工能耗。第 49~51 行更新机器的空闲能耗。相关符号及其解释如表 3 所示。

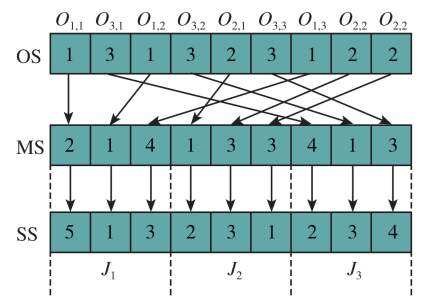


图 1 编码示意图

Fig. 1 Encoding Diagram

算法 1 Decodingprocess (OS,MS,SS)

Output : ($C_{\max}$, TEC)

01: $l_v = 0, e_v = 0, v = 1, 2, \dots, \zeta$

02: $j_i = 0, i = 1, 2, \dots, n$

03: $\omega_m = 1, A_m = \{(0, \text{int_max})\}, co_m = 0, spt_m = 0, m = 1, 2, \dots, \delta$

算法 1(续) Decodingprocess (OS,MS,SS)

```

04:  SPE=0, STE=0, SIE=0
05:  for  $k = 1$  to  $\sum_{i=1}^n n_i$  do
06:       $i \leftarrow OS[k]$ ,  $j_i \leftarrow j_i + 1$ ,  $\hat{m}(i, j_i) \leftarrow MS[\sum_{i'=1}^{i-1} n_i + j_i]$ ,  $\hat{s}(i, j_i) \leftarrow SS[\sum_{i'=1}^{i-1} n_i + j_i]$ 
07:       $T_{i, j_i, \hat{m}(i, j_i), \hat{s}(i, j_i)} = p_{i, j_i, \hat{m}(i, j_i), \hat{s}(i, j_i)} / R_s$ 
08:      SPE=SPE+  $T_{i, j_i, \hat{m}(i, j_i), \hat{s}(i, j_i)} \cdot \alpha_s$ 
09:       $spt_{\hat{m}(i, j_i)} = spt_{\hat{m}(i, j_i)} + T_{i, j_i, \hat{m}(i, j_i), \hat{s}(i, j_i)}$ 
10:  if  $j_i = 1$  then
11:       $\hat{v}(i, j_i) = \operatorname{argmin}_{v \in V} \{e_v + t_{l_v, 0}\}$ 
12:       $e_{\hat{v}(i, j_i)} = e_{\hat{v}(i, j_i)} + t_{l_{\hat{v}(i, j_i)}, 0} + t_{0, \hat{m}(i, j_i)}$ 
13:       $a_{i, j_i} = e_{\hat{v}(i, j_i)}$ 
14:      STE=STE+  $(t_{l_{\hat{v}(i, j_i)}, 0} + t_{0, \hat{m}(i, j_i)}) \cdot te$ 
15:  end if
16:  if  $j_i > 1$  then
17:      if  $\hat{m}(i, j_i - 1) \neq \hat{m}(i, j_i)$  then
18:           $\hat{v}(i, j_i) = \operatorname{argmin}_{v \in V} \{e_v + t_{l_v, \hat{m}(i, j_i - 1)}\}$ 
19:           $e_{\hat{v}(i, j_i)} = \max\{e_{\hat{v}(i, j_i)} + t_{l_{\hat{v}(i, j_i)}, \hat{m}(i, j_i - 1)} \cdot c_{i, j_i - 1}\} + t_{\hat{m}(i, j_i - 1), \hat{m}(i, j_i)}$ 
20:           $a_{i, j_i} = e_{\hat{v}(i, j_i)}$ 
21:          STE=STE+  $(t_{l_{\hat{v}(i, j_i)}, \hat{m}(i, j_i - 1)} + t_{\hat{m}(i, j_i - 1), \hat{m}(i, j_i)}) \cdot te$ 
22:      else
23:           $a_{i, j_i} = c_{i, j_i - 1}$ 
24:      end if
25:  end if
26:   $l_{\hat{v}(i, j_i)} = \hat{m}(i, j_i)$ 
27:  for  $q = 1$  to  $\omega_{\hat{m}(i, j_i)}$  do
28:      if  $t_{\hat{m}(i, j_i), 2q} - \max\{t_{\hat{m}(i, j_i), 2q-1}, a_{i, j_i}\} \geq T_{i, j_i, \hat{m}(i, j_i), \hat{s}(i, j_i)}$  then
29:           $c_{i, j_i} = \max\{t_{\hat{m}(i, j_i), 2q-1}, a_{i, j_i}\} + T_{i, j_i, \hat{m}(i, j_i), \hat{s}(i, j_i)}$ 
30:          if  $t_{\hat{m}(i, j_i), 2q-1} \geq a_{i, j_i}$  then
31:              if  $c_{i, j_i} = t_{\hat{m}(i, j_i), 2q}$  then
32:                  Delete the available time slots  $(t_{\hat{m}(i, j_i), 2q-1}, t_{\hat{m}(i, j_i), 2q})$  from  $A_{\hat{m}(i, j_i)}$ 
33:                   $\omega_{\hat{m}(i, j_i)} = \omega_{\hat{m}(i, j_i)} - 1$ 
34:              else
35:                   $t_{\hat{m}(i, j_i), 2q-1} = c_{i, j_i}$ 
36:              end if
37:          else
38:               $t_{\text{temp}} = t_{\hat{m}(i, j_i), 2q}$ ,  $t_{\hat{m}(i, j_i), 2q} = a_{i, j_i}$ 
39:              if  $c_{i, j_i} < t_{\text{temp}}$  then
40:                  Insert the available time slots  $(c_{i, j_i}, t_{\text{temp}})$  at position  $q + 1$  in  $A_{\hat{m}(i, j_i)}$ 
41:                   $\omega_{\hat{m}(i, j_i)} = \omega_{\hat{m}(i, j_i)} + 1$ 
42:              end if

```

算法 1(续) Decodingprocess (OS,MS,SS)

```

43:         end if
44:         break
45:     end if
46: end for
47:  $co_{\widehat{m}(i,j_i)} = \max\{co_{\widehat{m}(i,j_i)}, c_{i,j_i}\}$ 
48: end for
49: for  $b = 1$  to  $\delta$  do
50:     SIE = SIE +  $(co_b - spt_b) \cdot ie$ 
51: end for
52: TEC = SPE + SIE + STE
53:  $C_{\max} = \max_{i \in I} \{c_{i,n_i}\}$ 

```

表 3 算法 1 中的相关符号解释

Tab. 3 Explanation of Symbols in Algorithm 1

Symbols	Explanations
$\widehat{v}(i,j)$	运输工件 i 到机器加工工序 $O_{i,j}$ 的 AGV, $\widehat{v}(i,j) \in V$
j_i	工件 i 的工序编号
$\widehat{m}(i,j)$	被选中加工工序 $O_{i,j}$ 的机器, 表示为 $\widehat{m}(i,j) = \text{MS}[\sum_{i=1}^{i-1} n_i + j]$, 其中 $\widehat{m}(i,j) \in M_{i,j}$
e_v	AGV v 可以运送下一个工序的可用时间
l_v	AGV v 当前停靠的位置, 当 $l_v = 0$ 时表示 AGV v 停靠在装卸区, 而 $l_v = m (m = 1, 2, \dots, \delta)$, 表示 AGV v 停靠在机器 m 所在的位置
ω_m	机器 m 的可用时间段数量
$(t_{m,2q-1}, t_{m,2q})$	机器 m 的可用时间段从 $t_{m,2q-1}$ 到 $t_{m,2q}$, 其中 $q = \{1, 2, \dots, \omega\}$
A_m	机器 m 的可用时间段集合为 $A_m = \{(t_{m,1}, t_{m,2}), (t_{m,3}, t_{m,4}), \dots, (t_{m,2\omega-1}, t_{m,2\omega})\}$
spt_m	机器 m 上所有工序的总加工时间
SPE	所有工序的总加工能耗
SIE	所有机器的总空闲能耗
STE	所有工序的总运输能耗

2.2 四个子种群的初始化

在 MOCEA 中, 需要初始化三个进化种群和一个协助种群。协助种群是由 k 个完整的解(即, $(\pi_A^{ind}, \mu_A^{ind}, \tau_A^{ind})$)组成, 其中 π_A^{ind} , μ_A^{ind} 和 τ_A^{ind} 分别表示第 ind 个解的工序序列, 机器序列和速度序列。使用协助种群初始化三个进化种群。OS 种群包含 k 个工序序列, 以及对应协作者的索引。其中工序序列表示为 π^{ind} , 初始为 $\pi^{ind} = \pi_A^{ind}$; 其协作者索引表示为 $\varphi[ind]$, 初始为 $\varphi[ind] = ind$, $ind = 1, 2, \dots, k$ 。类似地, MS 种群包含 k 个机器序列, 以及对应协作者的索引。其中机器序列表示为 μ^{ind} , 初始为 $\mu^{ind} = \mu_A^{ind}$; 其协作者索引表示为 $\varphi[ind]$, 初始为 $\varphi[ind] = ind$, $ind = 1, 2, \dots, k$ 。同样地, SS 种群包含 k 个速度序列, 对应协作者的索引。其中速度序列表示为 τ^{ind} , 初始为 $\tau^{ind} = \tau_A^{ind}$; 其协作者索引表示为 $\psi[ind]$, 初始为 $\psi[ind] = ind$, $ind = 1, 2, \dots, k$ 。每个 π^{ind} , μ^{ind} 和 τ^{ind} 都与来自协助种群的协作者构成完整的解, 即 $(\pi^{ind}, \mu_A^{\varphi[ind]}, \tau_A^{\psi[ind]})$, $(\pi_A^{\varphi[ind]}, \mu^{ind}, \tau_A^{\psi[ind]})$ 和 $(\pi_A^{\psi[ind]}, \mu_A^{\varphi[ind]}, \tau^{ind})$ 。详细的初始化过程如算法 2 所示。

2.3 协同进化

2.3.1 OS 和 MS 中群进化过程。OS 和 MS 种群通过信息共享和协同进化机制进化, 并更新档案集。在 OS 种群进化过程中, 每个个体 π^{ind} 随机选择协助种群中的两个成员 $(\mu^{rand1}, \tau^{rand1})$ 进行协作, 构成完整解 $(\pi^{ind}, \mu^{rand1}, \tau^{rand1})$ 。同样, 在 MS 种群进化过程中, 每个个体 μ^{ind} 也会随机选择协助种群中的两个成员 π_A^{rand2} 和 τ_A^{rand2} 进行协作, 构成完整解 $(\pi_A^{rand2}, \mu^{ind}, \tau_A^{rand2})$ 。其中 $rand1$ 和 $rand2$ 是取自 $[1, k]$ 之间的随机整数。

算法 2 Population initialization**Output:** Assistant population, OS, MS, and SS populations

```

01:   for  $ind = 1$  to  $k$  do
02:        $\pi_A^{ind} \leftarrow$  random rule
03:        $\mu_A^{ind} \leftarrow$  random rule
04:        $\tau_A^{ind} \leftarrow$  random rule
05:        $(\pi_A^{ind}, \mu_A^{ind}, \tau_A^{ind}) \leftarrow$  Combination of  $\pi_A^{ind}$ ,  $\mu_A^{ind}$ , and  $\tau_A^{ind}$ 
06:        $\pi^{ind} \leftarrow \pi_A^{ind}$ ,  $\varphi[ind] = ind$ ,  $\mu^{ind} \leftarrow \mu_A^{ind}$ ,  $\varphi[ind] = ind$ ,  $\tau^{ind} \leftarrow \tau_A^{ind}$ ,  $\psi[ind] = ind$ 
07:   end for

```

为了更好地探索解空间并找到更优解,针对 OS 和 MS 中个体的不同特征,提出了两种基于关键路径的多邻域搜索策略,分别命名为 MNS1 和 MNS2。其中, MNS1 策略结合了按工序数分组的时间排序算子与基于关键路径的工序插入算子。其主要步骤描述如下: (1) 通过各工件的工序数进行分组。如图 2(a)所示,将序列分为 3 部分,即 G_1 , G_2 和 G_3 。(2) 从这些组中选择一个组 G_2 ,并按加工时间的降序对 G_2 中的两个工序 $O_{1,2}$ 和 $O_{3,2}$ 进行排序,其中 $O_{1,2}$ 的处理时间为 3, $O_{3,2}$ 的处理时间为 4,则得到 $[1, 3, 2, 3, 1, 1, 2, 3]$ 。(3) 从关键工序 $O_{1,3}$, $O_{3,1}$, $O_{3,2}$ 和 $O_{3,3}$ 中随机选择一个工序 $O_{1,3}$,并将其随机插入到其他位置,如图 2(b)所示。得到如图 2(c)所示的新序列 $[1, 3, 1, 2, 3, 1, 2, 3]$ 。MNS2 策略将均匀交叉(uniform cross-over, UC)操作与基于关键路径的变异算子相结合。

算法 3 Evolution process of OS and MS populations**Output:** New OS population, MS population, and new archive set

```

01:   for  $ind = 1$  to  $k$  do
02:        $rand1 \leftarrow$  Randomly choose an integer uniformly from the interval  $[1, k]$ 
03:        $(\pi^{ind}, \mu_A^{rand1}, \tau_A^{rand1}) \leftarrow$  Combination of  $\pi^{ind}$  and  $\mu_A^{rand1}$  with  $\tau_A^{rand1}$ 
04:        $\pi^{ind} \leftarrow$  Perform MNS1 to  $\pi^{ind}$ 
05:       if  $(\pi^{ind}, \mu_A^{rand1}, \tau_A^{rand1}) < (\pi_A^{ind}, \mu_A^{ind}, \tau_A^{ind})$  then
06:            $\pi^{ind} \leftarrow \pi^{ind}$ ,  $\varphi[ind] \leftarrow rand1$ 
07:       end if
08:       if  $(\pi^{ind}, \mu_A^{rand1}, \tau_A^{rand1}) < (\pi_A^{rand1}, \mu_A^{rand1}, \tau_A^{rand1})$  then
09:           update the archive set
10:       end if
11:   end for
12:   for  $ind = 1$  to  $k$  do
13:        $rand2 \leftarrow$  Randomly choose an integer uniformly from the interval  $[1, k]$ 
14:        $(\pi_A^{rand2}, \mu^{ind}, \tau_A^{rand2}) \leftarrow$  Combination of  $\mu^{ind}$  and  $\pi_A^{rand2}$  with  $\tau_A^{rand2}$ 
15:        $\mu^{ind} \leftarrow$  Perform MNS2 to  $\mu^{ind}$ 
16:       if  $(\pi_A^{rand2}, \mu^{ind}, \tau_A^{rand2}) < (\pi_A^{ind}, \mu^{ind}, \tau_A^{ind})$  then
17:            $\mu^{ind} \leftarrow \mu^{ind}$ ,  $\varphi[ind] \leftarrow rand2$ 
18:       end if
19:       if  $(\pi_A^{rand2}, \mu^{ind}, \tau_A^{rand2}) < (\pi_A^{rand2}, \mu_A^{rand2}, \tau_A^{rand2})$  then
20:           Update the archive set
21:       end if
22:   end for

```

在 UC 操作中:(1) 选择两个父代序列。(2) 为每个基因位置生成一个随机小数,形成一组小数 r 。(3) 对于每个基因,如果阈值(p)大于该位置的随机数(p 随机生成),则从第一个父代继承该位置的编号;否则,从第二个父代继承。(4) 关键路径上的变异操作则是在该路径上的工序,从其可用的机器集合中随机选择另一台机器。

MNS2 的具体过程如下:(1) 在 UC 操作中,从协助种群中随机选择一个机器序列(μ_A^4),并将其与当前个体(μ^3) 结合形成两个父本。(2) 生成一组 0 到 1 之间的随机小数 r ,其长度与 μ^3 相同。(3) 如果某位置的随机数 p 大于 r ,则新序列该位置的机器编号将继承自 μ^3 ;否则,将继承自 μ_A^4 的机器编号。具体过程如图 3(a)所示,得到新的机器序列(μ^3) 为[4, 2, 3, 2, 3, 1, 3, 2]。(4) 在关键路径上的加工机器上执行变异操作,以生成新序列。如图 3(b)所示,从关键工序 $O_{3,1}, O_{3,2}$ 和 $O_{3,3}$ 中选择一个工序,即 $O_{3,3}$ 。从其可用机器集合{2, 1}中随机选择另一台机器 1。(5) 因此,在图 3(c)中,机器序列从[4, 2, 3, 2, 3, 1, 3, 2]变为[4, 2, 3, 2, 3, 1, 3, 1]。这些策略充分考虑了个体的优势,增强了算法的全局搜索能力,同时在基于关键路径的操作中考虑了资源约束,对局部解进行探索。详细的 OS 和 MS 种群的进化过程如算法 3 所示。

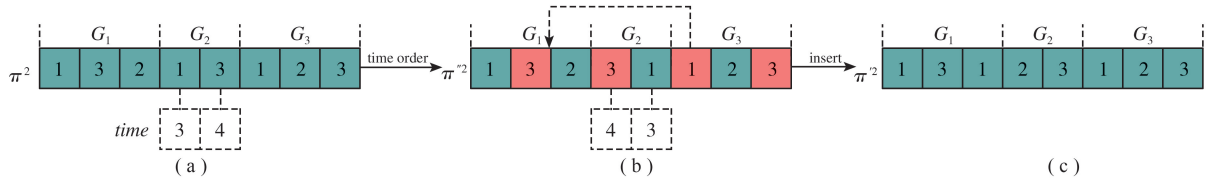


图2 MNS1 策略(红色为关键工序)

Fig. 2 MNS1 Strategy (Red Indicates Critical Operations)

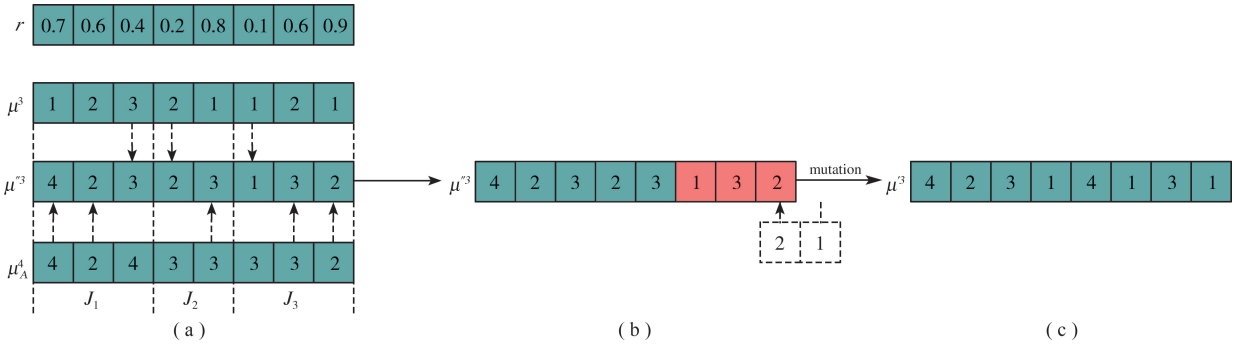


图3 MNS2 策略(红色为关键工序的加工机器)

Fig. 3 MNS2 Strategy (Red Indicates the Processing Machines for Critical Operations)

2.3.2 基于非关键路径的降耗策略的 SS 种群进化。与 OS 和 MS 种群进化过程类似,信息共享机制和协同进化机制也被应用到 SS 种群的进化过程中。本文从协助种群中随机选择一个工序序列和机器序列,并将其与当前的速度序列结合形成一个全新的解($\pi_A^{rand}, \mu_A^{rand}, \tau^{ind}$),其中 $rand$ 是 1 到 PS 之间的整数。

传统的关键路径策略通常忽略了非关键路径上的工序,然而,这些工序可能对整体的调度方案产生重要影响。因此,本文提出了一种结合了工序数分组的变异操作和基于非关键路径的降速操作多邻域搜索的能耗缩减策略(MNSER)。MNSER 的具体步骤如下:(1) 根据工序数将所有工序分组,如图 4(a)所示,分为 3 组。(2) 从这些组中随机选择一个组, G_2 。对于该组中的每个工序,其当前速度水平会被随机重新选择,随机选择的速度水平来自速度水平集合中的其他可用速度。(3) 构建解($\pi_A^{rand}, \mu_A^{rand}, \tau^{ind}$)的关键路径。(4) 在保持完工时间不变的情况下,尽量降低非关键路径上的速度水平。如图 4(c)所示,序列从[3, 1, 4, 4, 1, 1, 5, 4]变为[2, 1, 3, 4, 1, 1, 5, 4]。速度降低后的实际处理时间必须满足以下约束:

当前工序是工件的最后一道工序,也是机器上最后一道加工的工序。实际加工时间应小于或等于完工时间与当前工序的开始加工时间之间的差值,如式(32)所示,否则,实际加工时间应小于或等于当前工序的开始时间($st_{i,j}$)和同一台机器上后续工序的开始时间($st'_{i,j'}$)或下一个工序的负载开始时间($d_{i,j+1}$)之间

的最小值,如式(33)所示:

$$T_{i,j,m,s} \leq C_{\max} - st_{i,j}, \quad (32)$$

$$T_{i,j,m,s} \leq \min\{st'_{i,j'}, d_{i,j+1}\} - st_{i,j}. \quad (33)$$

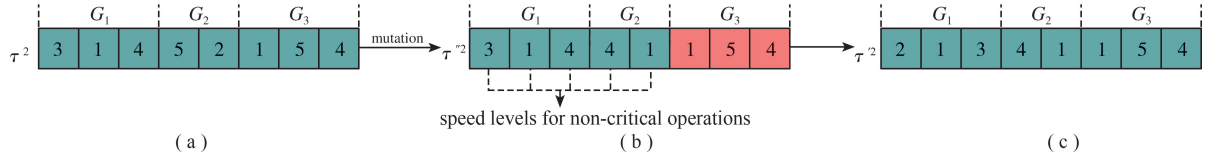


图4 MNSER 策略(红色为关键工序的速度水平)

Fig. 4 MNSER Strategy (Red Indicates the Speed Levels for Critical Operations)

算法4 Evolution process of SS population

Output: New SS population and new archive set

```

01:  for ind = 1 to k do
02:      rand ← Randomly choose an integer uniformly from the interval [1, k]
03:      ( $\pi_A^{rand}, \mu_A^{rand}, \tau^{ind}$ ) ← Combination of  $\tau^{ind}$  and  $\pi_A^{rand}$  with  $\mu_A^{rand}$ 
04:       $\tau^{ind}$  ← Perform MNSER to  $\tau^{ind}$ 
05:      if ( $\pi_A^{rand}, \mu_A^{rand}, \tau^{ind}$ ) < ( $\pi_A^{\psi[ind]}, \mu_A^{\psi[ind]}, \tau^{ind}$ ) then
06:           $\tau^{ind} \leftarrow \tau^{ind}, \psi[ind] \leftarrow rand$ 
07:      end if
08:      if ( $\pi_A^{rand}, \mu_A^{rand}, \tau^{ind}$ ) < ( $\pi_A^{rand}, \mu_A^{rand}, \tau^{rand}$ ) then
09:          Update the archive set
10:      end if
11:  end for

```

2.3.3 协助种群的更新。在每次迭代后,算法会生成许多优势解,这些解具备较高的适应性和多样性,涵盖了问题空间的不同区域和局部特征。为了进一步提升种群的质量,采用非支配排序对每次迭代生成的解进行排序,按照不同层次划分 Pareto 解,并选取最高层次的解,即 Pareto 前沿解。档案集的更新则通过新解与档案集中现有解的支配性判断来完成:如果新解支配了档案集中的某个解,则保留新解并删除被支配的解;若新解与档案集中的解互不支配,新解也会被保留。这一过程确保了档案集中的解不仅适应性强,而且多样性丰富,为后续的进化过程提供高质量的解集。在更新协助种群时,利用档案集中非支配解随机替换掉协助种群的解。当档案集中非支配解的数量少于协助种群大小时,随机选择与非支配解数量相等的协助种群解进行替换;如果非支配解数量大于协助种群大小,则随机选择等量的非支配解进行替换,以保持协助种群的适当规模。此更新过程确保了协助种群始终具有足够的多样性和适应性,有助于提高算法的全局搜索能力,促进算法的快速收敛并防止陷入局部最优。

在每次迭代后,算法会生成许多优势解,这些解具备较高的适应性和多样性,涵盖了问题空间的不同区域和局部特征。为了进一步提升种群的质量,采用非支配排序对每次迭代生成的解进行排序,按照不同层次划分 Pareto 解,并选取最高层次的解,即 Pareto 前沿解。档案集的更新则通过新解与档案集中现有解的支配性判断来完成:如果新解支配了档案集中的某个解,则保留新解并删除被支配的解;若新解与档案集中的解互不支配,新解也会被保留。这一过程确保了档案集中的解不仅适应性强,而且多样性丰富,为后续的进化过程提供高质量的解集。在更新协助种群时,利用档案集中非支配解随机替换掉协助种群的解。当档案集中非支配解的数量少于协助种群大小时,随机选择与非支配解数量相等的协助种群解进行替换;如果非支配解数量大于协助种群大小,则随机选择等量的非支配解进行替换,以保持协助种群的适当规模。此更新过程确保了协助种群始终具有足够的多样性和适应性,有助于提高算法的全局搜索能力,促进算法的快速收敛并防止陷入局部最优。

3 实验分析

3.1 实验设置

本文的所有仿真实验均在 Visual Studio 2022 中使用 C++ 实现。实验环境是一台具有 x64 架构的计算机,配备了 12 代 Intel © Core i7-12700 处理器,主频为 2.10 GHz,内存容量为 32.0 GB。选取了来自 <https://fastmanufacturingproject.wordpress.com/2019/04/11/fjspt-instances/> 的四组经典基准实例来评估算法的性能。这些基准实例分为以下四类:

基准 β_1 :由 Deroussi 和 Norre(2010)^[22] 提出的 10 个中规模实例组成。

基准 β_2 :由 Kumar 等人(2011)^[23] 提出的 28 个中规模实例组成。

基准 β_3 :由 Homayouni 等人(2023)^[24] 提出的 7 个大规模实例组成。

基准 β_4 :由 Homayouni 和 Fontes(2021)^[25] 提出的 20 个不同实例组成(包括 10 个小规模和 10 个中规模实例)。

表 4 显示了机器在不同速度下的单位加工能耗、机器的单位空闲能耗以及 AGV 的单位运输能耗。

表 4 单位能耗

Tab. 4 Unit Energy Consumption

S	R_s	α_s	te	ie
1	1.00			
2	1.30			
3	1.55	$4 \times (R_s)^2$	2.00	1.00
4	1.75			
5	2.10			

仿真实验参数设置中,将种群大小设置 k 为 50。每个算法针对每个实例独立运行了 20 次,运行时间为 $\xi \times 10^{-2}$ ms,其中 ξ 表示每个实例的总工序数。本文采用世代距离(Generational Distance, GD),反世代距离(Inverted Generational Distance, IGD)和集合覆盖率(Set Coverage, C)来评估算法的性能。

GD 用于衡量算法得到的 Pareto 前沿解的收敛性指标。它衡量了解集中每个解到 P^* 中的平均最小距离。GD 值越小,解集中的解就越接近 P^* ,算法的收敛性越好。具体如式(34)所示:

$$GD_k(E_k, P^*) = \frac{\sqrt{\sum_{x \in E_k} d(x, P^*)^2}}{|P^*|}, \quad (34)$$

式中 E_k 表示算法 k 获得的 Pareto 前沿, P^* 表示迄今为止某实例的最佳 Pareto 前沿。由于此问题的真实 Pareto 前沿未知,因此将所有算法获得的非支配解集合作为 P^* 的近似值。 $d(x, P^*)$ 表示 E_k 与 P^* 上所有解之间的最小欧几里得距离, $|P^*|$ 表示 P^* 的大小。

在此基础上,IGD 综合考量了 Pareto 前沿的收敛性和分布性,提供了多方面的性能评估,其计算由式(35)给出。较小的 IGD 值表示算法生成的 Pareto 前沿更接近真实的 Pareto 前沿,

$$IGD_k(E_k, P^*) = \frac{\sum_{x \in P^*} d(x, E_k)}{|P^*|}. \quad (35)$$

在计算 GD 和 IGD 时,考虑到两个目标的不同维度,本文进行了归一化处理,将数值范围缩放到 $[0, 1]$,以确保在多目标情况下的准确性。具体计算如式(36)所示:

$$f'_w(x) = \frac{f_w - f_w^{\min}}{f_w^{\max} - f_w^{\min}}, \quad (36)$$

式中 $f_w^{\min}$ 表示第 w 个目标的最小值, $f_w^{\max}$ 表示第 w 个目标的最大值。

C 指标衡量了两个解集之间的支配关系,计算如式(37)所示。C 值越大,表示算法 k 生成的 Pareto 前沿更优,其性能越好。具体而言,如果 $C(E_k, E_{k'})$ 的值为 1,则表示集合 $E_{k'}$ 中的所有解都被集合 E_k 中的某些解所支配。如果 $C(E_k, E_{k'})$ 的值为 0,则意味着集合 $E_{k'}$ 中的所有解都没有被集合 E_k 中的解所支配,

$$C(E_k, E_{k'}) = \frac{|\{x_1 \in E_{k'} \mid \exists x_2 \in E_k : x_2 \succ x_1\}|}{|E_{k'}|}。$$

(37)

3.2 与现有算法的比较

本节旨在验证 MOCEA 在求解多目标 FJSP-AGVs 问题时的有效性和优势。本文将 MOCEA 与最近发表的 EHA^[26]和 EMOEA^[27]算法进行了比较,以评估它们之间的性能差异。这两种算法在解决多目标 FJSP-AGVs 问题时展现出了不同的优势和特点,体现了良好的性能。通过与它们进行比较,本文能够更全面地评估 MOCEA 的性能。每种算法的参数设置都遵循了原始文献中的设置。表 5 左侧详细列出了三种算法在不同实例中的 GD 和 IGD 值,右侧列出了 C 指标的实验结果。

表 5 中的数据表明:(1) 在基于 65 个实例的统计分析中,MOCEA 在 GD 和 IGD 指标方面表现突出,分别在 52 和 51 个实例中取得最小值。(2) EHA 在 20 个实例中的 GD 和 16 个实例中的 IGD 表现最佳。(3) EMOEA 在 3 个实例中的 GD 和 1 个实例中的 IGD 中表现最佳。(4) 在 65 个实例中,MOCEA 的平均 GD 和 IGD 分别为 0.005 和 0.017。(5) 与 EHA 相比,MOCEA 的平均指标分别降低了 28.57%和 37.04%。同样,与 EMOEA 相比,这些指标分别降低了约 88.10%和 85.71%。(6) 就 C 指标而言,MOCEA 在 41 个实例中优于 EHA,在 65 个实例中优于 EMOEA。

图 5 展示了所有算法得到的 GD 和 IGD 指标的箱图。图 6 展示了 MOCEA,EHA 和 EMOEA 算法在所选实例(fjsp5,EXF21,Mk01 和 mjfs01)上得到的 Pareto 前沿图。图 5 可以看出,MOCEA 算法在大多数实例中表现出更低的 GD 和 IGD 值,这表明其生成的 Pareto 前沿解在收敛性和多样性方面都具有明显优势。此外,MOCEA 算法的箱体更为紧凑,显示出较低的结果波动性,这说明 MOCEA 在不同实验中的表现更加稳定。而 EHA 和 EMOEA 算法的箱体较大,尤其是在某些实例中波动显著,表明它们的解在不同实验中的离散性更高。极端值的对比也进一步表明,MOCEA 能够产生更接近最优解的结果,优于 EHA 和 EMOEA。

由图 6 可知,在大多数实例中,MOCEA 生成的 Pareto 前沿解不仅位于左下方,即更接近最优解,还在目标空间内分布更加均匀,显示出其在优化完工时间和能耗方面的优势。尤其是在 Mk01 实例中,MOCEA 的 Pareto 前沿解几乎完全支配了 EHA 和 EMOEA,体现了显著的性能优势。而在 EXF21 实例中,MOCEA 和 EHA 的解在某些目标上较为接近,但 MOCEA 的解在多样性上依然更具优势,表明其在权衡不同优化目标方面表现更为优异。这些图与数据表明,MOCEA 在求解多目标 FJSP-AGVs 问题时比其他算法具有明显优势。

表 5 MOCEA,EHA 和 EMOEA 的 GD,IGD 和 C 值
Tab. 5 GD, IGD, and C Values for MOCEA, EHA, and EMOEA

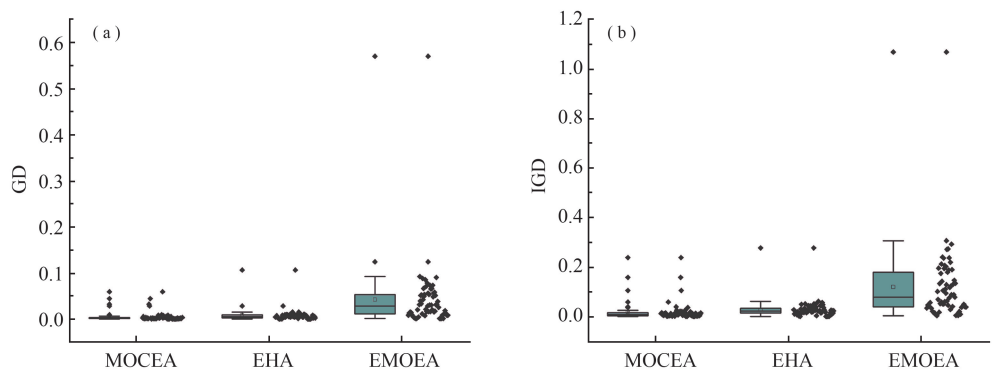
Instances	δ - n - ξ - ζ	MOCEA		EHA		EMOEA		C(MOCEA,	C(EHA,	C(MOCEA,	C(EMOEA,
		GD	IGD	GD	IGD	GD	IGD	EHA)	MOCEA)	EMOEA)	MOCEA)
β_1	fjsp1	0.002	0.018	0.005	0.051	0.056	0.128	0.333	0.297	1	0
	fjsp2	0.006	0.013	0.009	0.035	0.016	0.062	0.471	0.483	0.944	0.069
	fjsp3	0.007	0.02	0.007	0.042	0.028	0.094	0.200	0.556	1	0
	fjsp4	0.009	0.021	0.004	0.019	0.03	0.110	0.077	0.421	1	0
	fjsp5	0.003	0.005	0.008	0.025	0.016	0.045	0.706	0.333	1	0
	fjsp6	0.005	0.016	0.009	0.030	0.016	0.060	0.533	0.500	0.952	0
	fjsp7	0.004	0.006	0.010	0.038	0.021	0.075	0.533	0.296	1	0

表 5(续) MOCEA,EHA 和 EMOEA 的 GD,IGD 和 C 值
 Tab. 5(Continued) GD, IGD, and C Values for MOCEA, EHA, and EMOEA

Instances	δ - n - ξ - ζ	MOCEA		EHA		EMOEA		C(MOCEA, EHA)	C(EHA, MOCEA)	C(MOCEA, EMOEA)	C(EMOEA, MOCEA)	
		GD	IGD	GD	IGD	GD	IGD					
β_1	fjsp8	8-6-20-2	0.003	0.011	0.005	0.018	0.03	0.05	0.321	0.292	1	0
	fjsp9	8-5-17-2	0.003	0.016	0.001	0.003	0.007	0.028	0.240	0.667	0.929	0.100
	fjsp10	8-6-21-2	0.002	0.006	0.015	0.029	0.02	0.051	0.588	0.227	1	0
β_2	EXF11	4-5-15-2	0.005	0.006	0.015	0.056	0.078	0.271	0.765	0.294	1	0
	EXF12	4-5-15-2	0.002	0.022	0.008	0.043	0.049	0.179	0.735	0.32	1	0
	EXF13	4-5-15-2	0.006	0.025	0.009	0.022	0.053	0.194	0.333	0.591	1	0
	EXF14	4-5-15-2	0.008	0.025	0.008	0.032	0.071	0.202	0.519	0.350	1	0
	EXF21	4-6-18-2	0	0	0.011	0.061	0.073	0.273	1	0	1	0
	EXF22	4-6-18-2	0.003	0.039	0.004	0.039	0.048	0.206	0.333	0.360	1	0
	EXF23	4-6-18-2	0.008	0.01	0.009	0.021	0.085	0.188	0.421	0.444	1	0
	EXF24	4-6-18-2	0.003	0.015	0.010	0.044	0.065	0.237	0.652	0.200	1	0
	EXF41	4-5-15-2	0	0	0.009	0.024	0.042	0.088	0.960	0.071	1	0
	EXF42	4-5-15-2	0.001	0.001	0.005	0.019	0.015	0.058	0.810	0.185	1	0
	EXF43	4-5-15-2	0	0.036	0.005	0.044	0.011	0.040	1	0	0.556	0
	EXF44	4-5-15-2	0.001	0.007	0.005	0.022	0.017	0.052	0.786	0.211	0.933	0
	EXF51	4-5-15-2	0.001	0.005	0.004	0.014	0.033	0.105	0.625	0.125	1	0
	EXF52	4-5-15-2	0.001	0.007	0.008	0.015	0.031	0.117	0.567	0.231	1	0
	EXF53	4-5-15-2	0.004	0.012	0.004	0.016	0.039	0.142	0.440	0.533	1	0
	EXF54	4-5-15-2	0.001	0.003	0.009	0.019	0.053	0.123	0.667	0.095	1	0
	EXF71	4-8-24-2	0.005	0.011	0.01	0.056	0.066	0.236	0.500	0.214	1	0
	EXF72	4-8-24-2	0.001	0.001	0.009	0.051	0.066	0.174	0.667	0.125	1	0
	EXF73	4-8-24-2	0.001	0.019	0.010	0.048	0.073	0.219	0.542	0.200	1	0
	EXF74	4-8-24-2	0.001	0.004	0.009	0.033	0.088	0.240	0.533	0.125	1	0
	EXF81	4-6-18-2	0	0.005	0.008	0.036	0.056	0.189	0.909	0	1	0
	EXF82	4-6-18-2	0.002	0.012	0.004	0.016	0.032	0.133	0.286	0.370	1	0
	EXF83	4-6-18-2	0	0.003	0.006	0.028	0.052	0.130	0.727	0.150	1	0
	EXF84	4-6-18-2	0.001	0.004	0.009	0.025	0.058	0.195	0.5	0.290	1	0
	EXF91	4-5-15-2	0.001	0.012	0.007	0.022	0.051	0.146	0.789	0.200	1	0
	EXF92	4-5-15-2	0	0.003	0.004	0.026	0.031	0.088	0.818	0.118	1	0
	EXF93	4-5-15-2	0	0.001	0.005	0.017	0.038	0.101	0.875	0.057	1	0
	EXF94	4-5-15-2	0.002	0.003	0.003	0.016	0.051	0.134	0.880	0.143	1	0
β_3	Mk01	6-10-55-2	0	0	0.106	0.277	0.570	1.069	1	0	1	0
	Mk03	8-15-150-3	0.044	0.058	0	0	0.092	0.292	0	1	1	0
	Mk04	8-15-90-2	0	0	0.028	0.049	0.09	0.212	1	0	1	0
	Mk06	15-10-150-3	0.028	0.105	0	0	0.004	0.017	0	1	0	1
	Mk07	5-20-100-3	0	0.002	0.008	0.03	0.124	0.306	0.950	0	1	0
	Mk09	10-20-240-3	0.032	0.158	0	0	0.025	0.078	0	1	0	0.778
	Mk10	15-20-240-3	0.059	0.238	0	0	0.020	0.069	0	1	0	1

表 5(续) MOCEA,EHA 和 EMOEA 的 GD,IGD 和 C 值
Tab. 5(Continued) GD, IGD, and C Values for MOCEA, EHA, and EMOEA

Instances	δ - n - ξ - ζ	MOCEA		EHA		EMOEA		C(MOCEA,	C(EHA,	C(MOCEA,	C(EMOEA,
		GD	IGD	GD	IGD	GD	IGD	EHA)	MOCEA)	EMOEA)	MOCEA)
β_4	sfjs01	0.001	0.003	0	0.001	0.001	0.004	0.088	0.242	0.226	0.182
	sfjs02	0.001	0.001	0.001	0.003	0.001	0.003	0.120	0.040	0.120	0.04
	sfjs03	0.002	0.013	0	0.001	0.001	0.009	0.110	0.630	0.556	0.304
	sfjs04	0.002	0.007	0.001	0.002	0.001	0.007	0.127	0.627	0.408	0.390
	sfjs05	0	0.002	0	0	0.001	0.004	0.27	0.512	0.784	0.023
	sfjs06	0.001	0.014	0	0.014	0.002	0.014	0.254	0.500	0.667	0.125
	sfjs07	0.001	0.004	0.002	0.007	0.008	0.020	0.529	0.281	0.842	0.031
	sfjs08	0.002	0.011	0	0.019	0.003	0.016	0.121	0.700	0.721	0.167
	sfjs09	0.001	0.002	0.002	0.016	0.001	0.009	0.683	0.083	0.658	0.208
	sfjs10	0.003	0.012	0.001	0.007	0.005	0.021	0.279	0.452	0.733	0.095
	mfjs01	0.001	0.003	0.008	0.025	0.011	0.037	0.677	0.121	0.947	0.030
	mfjs02	0.001	0.004	0.003	0.012	0.009	0.033	0.690	0.257	0.968	0.057
	mfjs03	0.002	0.009	0.002	0.008	0.018	0.042	0.250	0.563	1	0
	mfjs04	0.001	0.006	0.003	0.02	0.014	0.056	0.625	0.161	1	0
	mfjs05	0.001	0.003	0.005	0.023	0.012	0.049	0.758	0.100	1	0
	mfjs06	0.001	0.007	0.007	0.013	0.015	0.042	0.550	0.276	0.962	0
	mfjs07	0.002	0.009	0.005	0.014	0.008	0.039	0.441	0.500	0.963	0
	mfjs08	0.005	0.015	0.005	0.01	0.013	0.046	0.444	0.478	1	0
	mfjs09	0.003	0.014	0.002	0.023	0.008	0.038	0.120	0.634	0.920	0
	mfjs10	0.002	0.012	0.004	0.027	0.008	0.034	0.500	0.261	0.895	0.174
Mean		0.005 9	0.017 3	0.007 9	0.027	0.042 6	0.119				



注:(a) GD 值;(b) IGD 值。

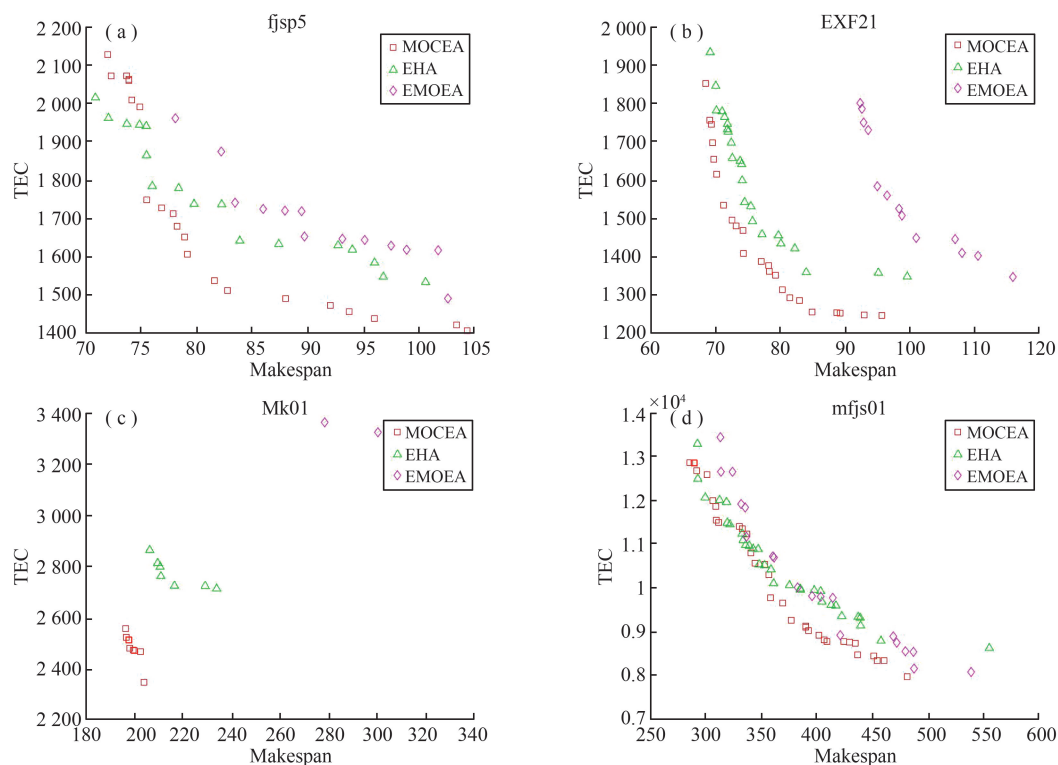
图 5 所有算法得到的箱图

Fig. 5 Boxplots of All Algorithms Obtained

3.3 Wilcoxon signed-rank 检验

本小节进一步验证了 MOCEA,EHA 和 EMOEA 之间的性能差异。对于 GD 和 IGD 指标,本文使用 Wilcoxon signed-rank 检验评估不同条件下 MOCEA 分别与对比算法之间的性能差异。在显著性水平 $\alpha = 0.05$ 下,如果计算得到的 $p-value$ 小于 α ,则表示算法之间存在显著性差异。表 6 展示了 Wilcoxon signed-rank 检验结果。结果表明 MOCEA 与 EHA 以及 MOCEA 与 EMOEA 在 GD 和 IGD 指标上的 $p-value$

$value$ 均为 0.000。因此,MOCEA 与其他算法之间的显著差异得到了进一步地验证。



注:(a) fjsp5; (b) EXF21; (c) Mk01; (d) mfjs01。

图6 所有算法在不同实例上的 Pareto 前沿

Fig. 6 Pareto Frontiers of All Algorithms on Different Instances

表6 所有实例的 Wilcoxon signed-rank 检验

Tab. 6 Wilcoxon Signed-Rank Test for All Instances

MOCEA vs.	GD			IGD		
	p -value	Significant ($p \leq 0.05$)	Decision	p -value	Significant ($p \leq 0.05$)	Decision
EHA	0.000	Yes	Reject H_0	0.000	Yes	Reject H_0
EMOEA	0.000	Yes	Reject H_0	0.000	Yes	Reject H_0

4 结论

绿色制造不仅可以降低能源消耗,而且还能促进经济的可持续发展。本文将可变加工速度这一约束引入到 FJSP-AGVs 问题中,构建了多目标 FJSP-AGVs 问题的数学模型。并采用一种多目标协同进化算法(MOCEA)求解该问题,针对问题特性设计了一系列进化策略。通过四组基准实例验证了所提算法的有效性。

未来研究可进一步考虑 AGV 数量和机器缓冲区大小对生产调度的影响,除此之外,更多实际生产中的约束也值得进一步研究。

参 考 文 献

- [1] LI Y, HUANG W, WU R, et al. An improved artificial bee colony algorithm for solving multi-objective low-carbon flexible job shop scheduling problem[J]. Applied Soft Computing, 2020, 95:106544.
- [2] 李俊青,李荣昊,陶昕瑞,等. 帝国竞争算法求解资源约束混合流水车间调度问题[J]. 聊城大学学报(自然科学版), 2022, 35(2): 14-26.
- [3] 秦浩翔,韩玉艳,李俊青,等. 求解阻塞混合流水车间调度的双层变异迭代贪婪算法[J]. 控制与决策, 2022, 37(9): 2323-2332.

- [4] TANG D B, DAI M. Energy-efficient approach to minimizing the energy consumption in an extended job-shop scheduling problem[J]. Chinese Journal of Mechanical Engineering, 2015, 28(5): 1048-1055.
- [5] GAO K, CAO Z, ZHANG L, et al. A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems[J]. Ieee-Caa Journal of Automatica Sinica, 2019, 6(4): 904-916.
- [6] GONG G, TANF J, HUANG D, et al. Energy-efficient flexible job shop scheduling problem considering discrete operation sequence flexibility[J]. Swarm and Evolutionary Computation, 2024, 84:101421.
- [7] TANG J, GONG G, PENG N, et al. An effective memetic algorithm for distributed flexible job shop scheduling problem considering integrated sequencing flexibility[J]. Expert Systems with Applications, 2024, 242: 122734.
- [8] PARK M, HAM A. Energy-aware flexible job shop scheduling under time-of-use pricing[J]. International Journal of Production Economics, 2022, 248: 108507.
- [9] LUO C, GONG W, LU C. Knowledge-driven two-stage memetic algorithm for energy-efficient flexible job shop scheduling with machine breakdowns[J]. Expert Systems with Applications, 2024, 235: 121149.
- [10] HE L, CHIONG R, LI W, et al. A multiobjective evolutionary algorithm for achieving energy efficiency in production environments integrated with multiple automated guided vehicles[J]. Knowledge-Based Systems, 2022, 243: 108315.
- [11] LI W, LI H, WANG Y, et al. Optimizing flexible job shop scheduling with automated guided vehicles using a multi-strategy-driven genetic algorithm[J]. Egyptian Informatics Journal, 2024, 25:100437.
- [12] HAN X, CHENG W, MENG L, et al. A dual population collaborative genetic algorithm for solving flexible job shop scheduling problem with AGV[J]. Swarm and Evolutionary Computation, 2024, 86: 101538.
- [13] ZHANG L, YAN Y, HU Y. Integrated scheduling of flexible job shop and energy-efficient automated guided vehicles[C] //2023 International Conference on Advanced Robotics and Mechatronics (ICARM), 2023: 493-498.
- [14] ZHOU X, WANG F, SHEN N, et al. A Green Flexible job-shop scheduling model for multiple AGVs considering carbon footprint[J]. Systems, 2023, 11(8): 427.
- [15] ZHANG M, WANG L, QIU F, et al. Dynamic scheduling for flexible job shop with insufficient transportation resources via graph neural network and deep reinforcement learning[J]. Computers & Industrial Engineering, 2023, 186: 109718.
- [16] PAN Q K, GAO L, WANG L. An effective cooperative co-evolutionary algorithm for distributed flowshop group scheduling problems [J]. IEEE Transactions On Cybernetics, 2022, 52(7): 5999-6012.
- [17] WANG Y H, HYAN Y, WANG Y, et al. An effective two-stage iterated greedy algorithm for distributed flowshop group scheduling problem with setup time[J]. Expert Systems with Applications, 2023, 233: 120909.
- [18] NIU W, LI J. A two-stage cooperative evolutionary algorithm for energy-efficient distributed group blocking flow shop with setup carryover in precast systems[J]. Knowledge-Based Systems, 2022, 257: 109890.
- [19] LI C, ZHANG B, HAN Y, et al. Energy-efficient hybrid flowshop scheduling with consistent sublots using an improved cooperative co-evolutionary algorithm[J]. Mathematics, 2022, 11(1): 77.
- [20] HE X, PAN Q K, GAO L, et al. A greedy cooperative co-evolutionary algorithm with problem-specific knowledge for multiobjective flowshop group scheduling problems[J]. IEEE Transactions On Evolutionary Computation, 2023, 27(3): 430-444.
- [21] 韩玉艳,李俊青,桑红燕,等. 离散 NSGA-II 求解带有限缓冲区的多目标批量流水线调度问题[J]. 聊城大学学报(自然科学版),2018,31(1):89-96.
- [22] DEROUSSI L, NORRE S. Simultaneous scheduling of machines and vehicles for the flexible job shop problem[C] //International conference on metaheuristics and nature inspired computing, Djerba Island, Tunisia, 2010: 1-2.
- [23] KUMAR M V S, JANARDHANA R, RAO C S P. Simultaneous scheduling of machines and vehicles in an FMS environment with alternative routing[J]. International Journal of Advanced Manufacturing Technology, 2011, 53: 339-351.
- [24] HOMAYOUNI S M, FONTES D B M M, GONCALVES J F. A multistart biased random key genetic algorithm for the flexible job shop scheduling problem with transportation[J]. International Transactions in Operational Research, 2023, 30(2): 688-716.
- [25] HOMAYOUNI S M, FONTES D B M M. Production and transport scheduling in flexible job shop manufacturing systems[J]. Journal of Global Optimization, 2021, 79(2): 463-502.
- [26] XU G, BAO Q, ZHANG H. Multi-objective green scheduling of integrated flexible job shop and automated guided vehicles[J]. Engineering Applications of Artificial Intelligence, 2023, 126: 106864.
- [27] WEN X, FU Y, YANG W, et al. An effective hybrid algorithm for joint scheduling of machines and AGVs in flexible job shop[J]. Measurement and Control, 2023, 56(9-10): 1582-1598.

(责任编辑:刘庆松)