

基于均衡机器能耗的绿色柔性作业车间多目标调度*

蒲虹羽¹, 马冉¹, 张玉忠²

(1. 青岛理工大学管理工程学院, 山东青岛 266520; 2. 曲阜师范大学运筹学研究院, 山东日照 273165)

摘要:在实际的车间生产系统中,机器是生产过程中不可或缺的一部分。机器若长期处于高能耗状态,会加速设备老化,增加故障发生概率,从而扰乱生产周期。相反,机器若长期处于闲置或低能耗状态,则会造成资源严重浪费。因此,能耗平衡是一个很值得探索的问题。针对上述问题,对均衡机器能耗的绿色柔性作业车间调度问题(green flexible job shop scheduling problem with balanced machine energy consumption, GFJSP-BMEC)进行探讨,构建了最小化最大完工时间、机器间能耗差异与总能耗的加权和的双目标优化模型,提出了改进的第2代非支配排序遗传算法(improved non-dominated sorting genetic algorithm II, INSGA-II),并通过大量的数值实验证明了均衡机器能耗策略对完工时间和总能耗的影响。将 INSGA-II 与第2代非支配排序遗传算法和多目标粒子群算法进行了比较,证明 INSGA-II 在求解 GFJSP-BMEC 时的有效性和优越性。

关键词:绿色调度;多目标优化;柔性作业车间;均衡能耗;INSGA-II

中图分类号:O224

文献标志码:A

文章编号:1672-6693(2026)01-0007-20

根据国际能源署的统计数据,全球能源消耗预计将从2010年的383 EJ增长至2050年的536 EJ,增幅约40%;作为能源消耗的主要领域之一,制造业的能源消耗量占全球能源消耗总量的38%,二氧化碳排放量占全球二氧化碳排放总量的47%^[1]。在此背景下,各国纷纷出台政策推动制造业低碳转型发展:中国提出“双碳”战略目标,要求工业部门到2030年实现碳达峰;欧盟实施碳边界调整机制,通过征收碳关税迫使制造业减排;美国的《降低通货膨胀法案》则设立了3690亿美元的清洁技术补贴,以加快制造业的绿色技术创新。在这些政策的推动下,全球制造业正加速向绿色制造转型,且这一趋势已成为行业发展的重要方向。绿色制造综合考虑了经济影响、资源利用、环境影响等诸多因素,贯穿产品的全生命周期,包括设计、制造、包装、运输、使用、废弃和再制造过程,旨在实现经济与社会的和谐发展^[2]。

制造业进行绿色转型的有效手段是对生产过程进行优化调度^[3],而离散制造作为制造业的两大基本范式之一,在推动制造业绿色转型中具有战略支点作用;其中,作业车间是离散制造领域的典型生产模式。在传统的作业车间调度问题(job shop scheduling problem, JSP)中,每个工件由1组具有固定加工顺序的工序组成,且每道工序必须分配给1台特定机器加工,不可替换^[4],该问题已被证明是非确定性多项式时间难问题(non-deterministic polynomial-time hard, NP难问题)^[5]。近些年来,随着工业的发展,柔性作业车间被广泛应用于多品种、小批量、频繁订单插入等复杂场景^[6],相较于传统作业车间在JSP设备分配上的刚性约束,柔性作业车间允许工序在可选机器集合中选择1台机器进行加工。这在传统作业车间基础上增加了调度复杂性^[7],但更贴近实际生产情况,因而柔性作业车间更适合制造业智能升级和绿色转型需求。

柔性作业车间调度问题(flexible job shop scheduling problem, FJSP)是一个NP难问题^[8],它的优化指标通常与加工时间、加工质量、成本、资源利用率等经济指标密切相关^[3]。在单目标FJSP中,与时间相关的目标是被最广泛采用的优化指标。为了最小化完工时间,潘全科等人^[9]采用遗传算法求解FJSP;Yazdani等人^[10]针对FJSP提出了变邻域搜索算法;Yuan等人^[11]提出了混合微分进化算法来求解FJSP。此外,Zeng等人^[12]研究了以最小化总完工时间为目标具有多个流程计划的FJSP;鲍蕾等人^[13]也提出了改进的混合蛙跳算法来解决

* 收稿日期:2025-07-14 修回日期:2025-10-30 网络出版时间:2026-03-03T14:15

资助项目:国家自然科学基金面上项目(No. 12271295, No. 12371319);山东自然科学基金面上项目(No. ZR2024MA026);山东自然科学基金面上项目(No. 2025MS102)

第一作者简介:蒲虹羽,女,研究方向为工程调度优化,E-mail:phylucky@163.com;通信作者简介:马冉,女,教授,E-mail:sungirlmr@126.com

网络出版地址:https://link.cnki.net/urlid/50.1165.N.20260303.1044.006

FJSP, 目标是 minimized 最大完工时间; Zhang 等人^[14]也提出了海洋捕食者算法来优化最大完工时间。在多目标 FJSP 研究中, 通常将时间与成本、质量、资源利用率等目标结合。Shen 等人^[15]采用改进的禁忌搜索算法降低总能耗成本和最大完工时间; Lou 等人^[16]提出了一种基于学习和分解的多目标记忆算法来解决考虑了人为因素的 FJSP 问题, 目标是 minimized 最大完工时间、机器负载和机器总负载; Vital-Soto 等人^[17]使用带有创新算子的第 2 代非支配排序遗传算法(non-dominated sorting genetic algorithm II, NSGA-II), 最大限度地减少了工期、最大工人工作量和加权拖期; Sun 等人^[18]提出带有禁忌搜索的混合非支配排序遗传算法, 在最大完工时间、总拖期和机器切换次数之间进行权衡; Wei 等人^[19]建立了一个供需匹配模型, 目标是 minimized 总成本和最大完工时间。

然而, 随着制造业绿色转型的推进, 如何在传统制造系统向绿色制造系统转型的过程中实现生态与经济效益双赢成为行业发展的关键课题, 在此背景下, 绿色柔性作业车间调度问题(green flexible job shop scheduling problem, GFJSP)引起了学者的广泛关注。有关 GFJSP 的研究可分为 2 类: 第 1 类是以能耗为优化目标的研究。张洪亮等人^[20]提出了一种改进的非支配排序遗传算法, 以 minimized 最大完工时间和总能耗; Chen 等人^[21]提出了一种结合 Q 学习的多目标免疫算法来求解三目标的 FJSP, 旨在 minimized 最大完工时间、总能耗和平均一致性指数; Zhang 等人^[22]旨在优化 FJSP 的生产周期、系统可用性和总能耗; Xiao 等人^[23]建立了一个模型, 目标是 minimized 最大完工时间、刀具数量、机器负载和机器能耗; 蒋权等人^[24]提出了一种多目标近端策略优化算法来求解以 minimized 生产间隔、拖期和加工能耗为目标的模型; 江雨燕等人^[25]构建了一个多目标调度优化模型, 目标是 minimized 最大完工时间、总能耗和总机器负载; Zhang 等人^[26]设计了一种改进的记忆算法, 可同时优化最大完工时间、总能耗和机器负载平衡。第 2 类是与环境影响(如噪音、废水和碳排放)相关优化目标的研究。Liu 等人^[27]提出了一个旨在减少产品碳足迹的模型, 该模型采用 NSGA-II 进行求解; 李益兵等人^[28]基于碳排放、噪音和废物共 3 个指标对环境污染进行了评估, 并提出了一种改进的人工蜂群算法, 来最大限度地减少最大完工时间和环境污染; Wang 等人^[29]提出了一种新型多目标进化算法, 采用 3 种改进策略, 以 minimized 完工时间、总拖期、加工负载率和总碳排放量为优化目标。

上述研究的异同详见表 1。综上所述, 尽管现有的研究从不同角度对 GFJSP 展开探讨, 但现有能耗优化研究主要聚焦于降低生产系统的总能耗, 对单个机器间能耗均衡的关注相对有限, 机器间能耗的差异可能带来一系列问题。具体来说, 某些机器可能长期处于高能耗状态, 这会加速设备损耗, 增加机器故障概率, 甚至扰乱生产计划; 相反, 其他机器若处于闲置状态或以低能耗水平运行, 则会造成资源浪费。因此, 在 GFJSP 中引入机器能耗均衡机制具有重要的理论和实践价值。

此外, 在实际生产过程中, 工期、总能耗和均衡机器能这 3 个目标之间还存在耦合关系, 主要体现在以下 3 个方面: 第一是时间和总能耗之间的耦合关系^[30]。提高机器功率可以缩短加工时间, 但同时会导致能耗上升。第二是时间和机器能耗均衡之间的耦合关系^[31]。为实现能耗均衡而选择低能耗机器加工时, 往往可能需要更长的加工时间, 从而导致最大完工时间增加。第三是总能耗与机器能耗均衡的耦合关系。为了均衡机器之间的能耗, 可能需要避免某些机器长时间闲置, 这可能会导致某些能耗较高的机器加工更多工件, 从而增加总能耗。因此, 为有效处理这些耦合关系并降低计算复杂度, 本文建立了一个双目标优化模型, 一方面 minimized 最大完工时间, 另一方面 minimized 总能耗与机器间能耗差异的加权和。使用改进的第 2 代非支配排序遗传算法(improved non-dominated sorting genetic algorithm II, INSGA-II)同时优化 2 个目标, 并通过数据模拟不断调整权重系数 α , 找到一个合理的取值, 以同时优化总能耗和机器能耗差异这 2 个子目标。最后, 通过大量数值实验证明了该算法在求解均衡机器能耗的绿色柔性作业车间调度问题(green flexible job shop scheduling problem with balanced machine energy consumption, GFJSP-BMEC)时的优势。

本文的创新点主要有以下几个方面:

- 1) 创新性地引入机器能耗均衡优化指标。本文提出了均衡机器能耗优化目标, 旨在解决生产过程中某些机器长期处于高耗能状态导致故障概率增加, 而其他机器却处于闲置状态造成资源浪费的问题, 提高生产系统的资源利用率, 推动绿色生产。

- 2) 在模型中将均衡机器能耗与总能耗以加权形式结合起来。该模型不仅能平衡各机器的能耗, 还能兼顾整体能耗控制, 因此更符合实际生产情况, 有助于更有效地实现绿色生产。

- 3) 提出 INSGA-II。在 NSGA-II 的基础上改进了解码方法, 采用基于均衡机器能耗的插入式贪婪解码方法。首先, 这种解码方法有助于减少工序间隔数, 提高机器利用率。其次, 它可以与权重 α 的调整策略相结合,

更好地同时优化总能耗和不同机器之间的能耗差异这 2 个子目标。最后,它加快了算法的收敛速度。

4) 开展多组数值实验验证模型与算法的有效性。一方面,通过调整决定总能耗和机器间能耗差异相对重要性的权重系数 α ,分析了均衡能耗策略对最大完工时间和总能耗的影响,并给出了 α 的合理取值范围;另一方面,通过比较 INSGA-II 和其他算法,证明了 INSGA-II 在解决所提问题方面的优越性。

表 1 FJSP 和 GFJSP 的相关研究
Tab. 1 Related research of FJSP and GFJSP

类别	作者	优化目标				
		时间相关目标	其他经济目标	能耗目标	环境影响目标 (噪音、碳排放等)	其他目标
单目标 FJSP	潘全科等人 ^[9]	加工周期				
	Amiri 等人 ^[10]	完工时间				
	Yuan 等人 ^[11]	完工时间				
	Zeng 等人 ^[12]	总完工时间				
	鲍蕾等人 ^[13]	最大完工时间				
	Zhang 等人 ^[14]	最大完工时间				
双目标和多 目标 FJSP	Shen 等人 ^[15]	最大完工时间	总能耗成本			
	Lou 等人 ^[16]	最大完工时间	机器负载、总机 器负载			
	Vital-Soto 等人 ^[17]	最大完工时间、加权 拖期	工人工作量			
	Sun 等人 ^[18]	最大总完工时间、加 权拖期	机器切换次数			
	Wei 等人 ^[19]	最大完工时间、拖期	总成本			
最小化资源 消耗 GFJSP	张洪亮等人 ^[20]	最大完工时间		总能耗		
	Chen 等人 ^[21]	最大完工时间		总能耗		平均一致性指数
	Zhang 等人 ^[22]	最大完工时间		总能耗		系统可用性
	Xiao 等人 ^[23]	最大完工时间	工具质量、机器 负载	机器能耗		
	蒋权等人 ^[24]	最大完工时间、拖期		加工能耗		
	江雨燕等人 ^[25]	最大完工时间	机器总负载	总能耗		
	Zhang 等人 ^[26]	最大完工时间	机器负载均衡	总能耗		
最小化环境 影响 GFJSP	Liu 等人 ^[27]				产品碳足迹	
	李益兵等人 ^[28]	最大完工时间			环境污染	
	Wang 等人 ^[29]	完工时间、总拖期	加工负荷率	总碳排放		

本文的剩余部分结构概述如下:第 1 节介绍了 GFJSP-BMEC 并构建了相应的模型;第 2 节详细说明了 INSGA-II;第 3 节验证了新目标对最大完工时间和总能耗的影响,并通过数值实验证明了 INSGA-II 在求解 GFJSP-BMEC 时的有效性;第 4 节总结了研究结果,并展望未来研究方向。

1 问题描述和模型

1.1 问题描述

GFJSP-BMEC 可描述如下,有 m 台机器的集合 $M(M = \{M_1, M_2, \dots, M_m\})$ 和 n 个工件的集合 $J(J = \{J_1, J_2, \dots, J_n\})$,每台机器具有一个加工能耗 P_{ijk} 和空闲能耗 P_{sk} ,每个工件 J_i 由 r 道具有优先顺序约束的工序 N_i

组成, $N_i = \{O_{i1}, O_{i2}, \dots, O_{ir}\}$, 每道工序 O_{ij} 在可选的机器集合 M_{ij} 中选择一台机器进行加工, 不同工序在不同机器上的加工时间 T_{ijk} 存在差异。GFJSP-BMEC 的任务就是将每道工序分配给一台机器, 并确定它们的加工顺序。

表 2 给出 GFJSP-BMEC 的一个实例: 有 3 台机器 M_1, M_2, M_3 和 4 个工件 J_1, J_2, J_3, J_4 , 且 J_1 有 3 道工序 O_{11}, O_{12}, O_{13} ; J_2 有 2 道工序 O_{21}, O_{22} ; J_3 有 3 道工序 O_{31}, O_{32}, O_{33} ; J_4 有 2 道工序 O_{41}, O_{42} 。其中 O_{11} 可以在 M_1 或 M_2 上加工, M_1 的加工时间为 10 个单位, M_2 的加工时间为 15 个单位。表 3 列出了机器的参数, M_1 的待机功率和加工功率分别为 0.4 和 4.5 kW。

表 2 GFJSP-BMEC 实例
Tab. 2 An instance of GFJSP-BMEC

工件	工序	加工时间		
		M_1	M_2	M_3
J_1	O_{11}	10	15	
	O_{12}		20	25
	O_{13}	15		30
J_2	O_{21}	12		14
	O_{22}		18	20
J_3	O_{31}	10	15	
	O_{32}	13		15
	O_{33}		21	23
J_4	O_{41}	5		11
	O_{42}	14	16	

表 3 机器参数
Tab. 3 Machine parameters

机器	待机能耗/kW	加工能耗/kW
M_1	0.4	4.5
M_2	0.5	5.8
M_3	0.6	5.3

1.2 模型假设和约束

根据实际生产和加工条件作出以下假设:

- 1) 每台机器最多只能同时加工 1 道工序;
- 2) 每道工序只能在 1 台机器上进行加工;
- 3) 机器之间相互独立;
- 4) 同一工件中各操作的加工顺序遵循优先级约束;
- 5) 机器设置时间可忽略不计;
- 6) 工序的加工时间是预先确定的;
- 7) 工序一旦在机器上开始加工就不能中断。

1.3 符号

本文使用的符号如表 4 所示。

1.4 数学模型

本文以 GFJSP-BMEC 为研究对象, 旨在最小化最大完工时间 f_1 以及机器间能耗差和总能耗的加权和 f_2 , 机器间能耗差 f'_1 和总能耗 f'_2 是构成 f_2 的 2 个子目标, 因此可以建立如下模型。

最小化最大完工时间 f_1 :

$$\min f_1 = \min C_{\max} = \min\{\max_{i=1, \dots, n} C_i\}。$$

最小化机器之间的能耗差异 f'_1 :

$$\min f'_1 = \min E_v = \min \frac{1}{m} \sum_{k=1}^m (E_k - \bar{E})^2.$$

其中:

$$\bar{E} = \frac{1}{m} \sum_{k=1}^m E_k,$$

$$E_k = \sum_{i=1}^n \sum_{j=1}^r X_{ijk} T_{ijk} P_{ijk} + \sum_{l=1}^s P_{sk} T_{sk}, \forall M_k \in M.$$

最小化总能耗 f'_2 :

$$\min f'_2 = \min E = \min \left(\sum_{i=1}^n \sum_{j=1}^r \sum_{k=1}^m X_{ijk} P_{ijk} + \sum_{k=1}^m \sum_{l=1}^s P_{sk} T_{sk} \right).$$

表 4 相关符号说明

Tab. 4 Explanation of relevant symbols

变量类型	符号	含义	变量类型	符号	含义
索引	J	工件索引, $J = \{J_1, J_2, \dots, J_i, \dots, J_n\}$	参数	C_i	工件 i 的完工时间
	N_i	J_i 的工序索引, $N_i = \{O_{i1}, O_{i2}, \dots, O_{ir}\}$		P_{sk}	机器 k 的待机能耗
	M	机器索引, $M = \{M_1, M_2, \dots, M_m\}$		T_{sk}	机器 k 的待机时间
参数	n	工序的数量	决策变量	E_v	所有机器的能耗差异
	r	工件 i 工序的数量		E_k	机器 k 的能耗
	m	机器数量		E	所有机器的总能耗
	s	相邻工序之间间隔的数量		\bar{E}	所有机器的平均能耗
	O_{ij}	工件 i 的第 j 道工序		α	权重系数, 调节 E 和 E_v 的相对重要性, $0 \leq \alpha \leq 1$
	M_{ij}	O_{ij} 可选的机器集合		X_{ijk}	如果工序 O_{ij} 在机器 k 上加工, $X_{ijk} = 1$; 否则 $X_{ijk} = 0$
	S_{ijk}	工序 O_{ij} 在机器 k 上的开始时间		Y_{ijghk}	如果在机器 k 上工序 O_{ij} 在 O_{gh} 前加工, $Y_{ijghk} = 1$; 否则 $Y_{ijghk} = 0$
	T_{ijk}	工序 O_{ij} 在机器 k 上的加工时间			
	C_{ijk}	工序 O_{ij} 在机器 k 上的完工时间			
	P_{ijk}	在机器 k 上加工 O_{ij} 的能耗			

为了解决目标间耦合问题并简化求解过程,本研究引入权重系数 α 将 f'_1 和 f'_2 整合为 1 个目标函数。当 $\alpha=1$ 时,完全侧重于总能耗的最小化;当 $\alpha=0$ 时,完全聚焦于均衡各机器的能耗;当 $0 < \alpha < 1$ 时,将在总能耗和机器间的能耗均衡 2 个子目标上实现协同优化。

最小化机器间能耗差和总能耗的加权和 f_2 :

$$\min f_2 = \min E = \min(\alpha E_v + (1-\alpha)E),$$

$$\text{s. t. } \sum_{k \in M_{ij}} X_{ijk} = 1, \tag{1}$$

$$\sum_{i=1}^n \sum_{j=1}^r \sum_{k=1}^m Y_{ijghk} \leq 1, \tag{2}$$

$$\sum_{g=1}^n \sum_{h=1}^r \sum_{k=1}^m Y_{ijghk} \leq 1, \tag{3}$$

$$C_{ijk} = S_{ijk} + T_{ijk}, \tag{4}$$

$$S_{ghk} \geq (S_{ijk} + T_{ijk}) \times Y_{ijghk}, \tag{5}$$

$$S_{i(j+1)k} \geq C_{ijk}. \tag{6}$$

其中:式(1)是工序分配的约束,确保每道工序只分配给可选机器集合 M_{ij} 中的一台机器加工;约束条件(2)和(3)规定每道工序最多只能有 1 道紧前工序或紧后工序;式(4)定义了每道工序的完工时间约束;约束条件(5)是

同一台机器上工序的加工顺序约束;约束条件(6)保证在同一工件中工序要按照先后顺序进行加工。

2 INSGA- II

2.1 算法框架

FJSP 是一个 NP 难问题^[5],通常采用启发式和元启发式算法来获取近似最优解。在多目标 FJSP 优化领域,NSGA- II 因具有快速的搜索能力和良好的鲁棒性而被证实是一种高效的求解方法^[32]并得到广泛应用。对于本文研究的 GFJSP-BMEC,在 NSGA- II 的基础上进行了针对性改进,提出 INSGA- II,具体操作流程如下。

第 1 步,设置参数。本文设置了迭代次数 G_{max} 、交叉概率 P_c 、变异概率 P_m 。

第 2 步,初始化。随机生成初始父种群 P_t ,种群数量为 N 。

第 3 步,快速非支配排序。根据目标函数值对当前种群进行排序,然后再划分为不同的非支配前沿。

第 4 步,拥挤度计算。计算每个非支配前沿中个体的拥挤距离。

第 5 步,锦标赛选择。根据非支配前沿和拥挤度进行选择操作,优先选择非支配层级较低的个体,如果个体在同一非支配前沿内,则选择拥挤度较大的个体。

第 6 步,交叉和变异。分别对工序(operation sequence, OS)层和机器(machine selection, MS)层进行交叉和变异操作,生成新种群。

第 7 步,精英保留策略。合并父代种群和子代种群,执行快速非支配排序和拥挤度计算,保留最优的 N 个个体作为新的父代种群 P_{t+1} 。

第 8 步,终止条件检查。如果达到预定的迭代次数,算法终止,执行第 9 步,否则返回第 5 步。

第 9 步,输出结果。将最后一代种群中的非支配解作为最优解输出。

图 1 进一步展示了 INSGA- II 的流程。

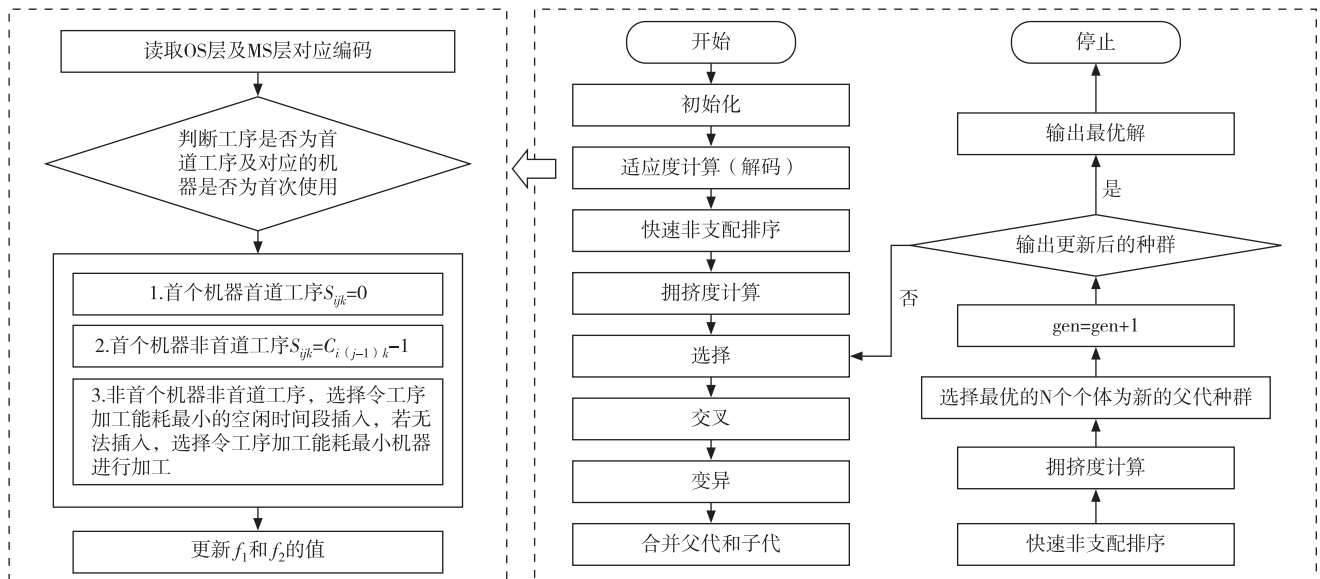


图 1 INSGA- II 流程图

Fig. 1 Flowchart of INSGA- II

2.2 编码和解码

2.2.1 编码

本文采用双层编码方式,包括 OS 层和 MS 层共 2 个部分,且 2 层染色体长度相同^[22]。在 OS 层编码中,染色体基因数等于总工序数,每道工序由相应的工件编号表示,而工件编号出现的次数对应它的工序数,从左至右扫描 OS 层染色体时,某个工件编号的第 k 次出现即代表该工件的第 k 道工序;与之对应的 MS 层编码按照工序顺序排列,每个基因位置从对应工序的可选机器集合中随机选择 1 台,构建出 MS 层染色体。以表 2 中的实例为例,图 2 为该实例的编码方法。OS 的染色体编码 $\{3, 2, 3, 1, 4, 3, 2, 1, 4, 1\}$ 表示按工序 $(O_{31}, O_{21}, O_{32}, O_{11}, O_{41}, O_{33}, O_{12}, O_{41}, O_{13})$ 的顺序加工;MS 层从左到右按工件工序排列,数值表示对应工序所选的加工机器,以 OS 层第 1 基因位的 3 为例,此次加工工序为 O_{31} ,选择机器 M_2 加工,其他工序同理。

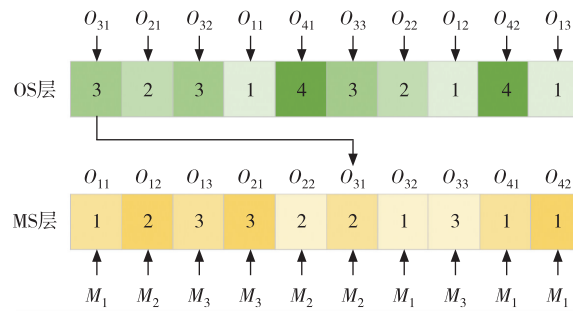


图 2 编码示例

Fig. 2 Example for encoding

2.2.2 解码

解码是将编码转化成具体调度方案的过程,考虑到绿色指标的优化,本文采用了一种基于能耗的插入式贪婪解码算法,具体解码过程如下。

从左到右依次读取 OS 层及对应 MS 层编码,判断读取到的工序 O_{ij} 是否是工件的首道工序,以及对应机器是否是首次使用,若工序 O_{ij} 是工件 i 的首道工序,且对应的加工机器 k 是首次使用,则将 0 作为工序 O_{ij} 的起始加工时间;若工序非工件 i 的首道工序,但对对应机器 k 是首次使用,则将工序 $O_{i(j-1)}$ 的加工结束时间作为工序 O_{ij} 的起始加工时间;若工序非工件 i 的首道工序且若工序 O_{ij} 对应机器 k 并非首次使用,依次检查此工序所有可选机器上的空闲时间段,选择令工序加工能耗最小的空闲时间段插入,若无法插入,选择令工序加工能耗最小机器进行加工,最后更新 f_1 和 f_2 的值。

2.3 快速非支配排序

快速非支配排序是一个根据个体在群体中的支配关系对个体进行排序的系统过程。具体步骤如下:

第 1 步,初始化参数。每个个体 s 设定 2 个参数: $n(s)$ 表示支配个体 s 的解的数量, $g(s)$ 表示被个体 s 支配的解集合。

第 2 步,找到第 1 层非支配个体。在种群中寻找 $n(s)=0$ 的个体,并将它们存入集合 F_1 ,这些个体组成了第 1 层非支配前沿,为其中的个体赋予相同的非支配等级 $R=1$ 。

第 3 步,更新支配关系。对于集合 F_1 中的每个个体 t ,查看它支配的个体集合 $g(t)$ 。将集合 $g(t)$ 中的每个个体 k 的 $n(k)$ 值减去 1(因为已经有 1 个支配它的个体被放入了 F_1)。

第 4 步,处理下一级非支配个体。若 $n(k)-1=0$,将它放入集合 F_2 ,并赋予这些个体相应的非支配等级 $R=2$ 。

第 5 步,递归分级。以此类推,直到所有个体都被分级完成。

如图 3 所示,对解 f 而言,解 f 被 a 、 b 和 c 这 3 个解支配,所以 $n(c)=3$ 。对解 b 而言,解 b 支配 d 、 e 、 f 和 g ,因此, $g(b)=\{d, e, f, g\}$ 。

2.4 拥挤度计算

通过拥挤度计算,算法能够在非支配排序后的同一非支配前沿的个体中进一步区分优劣,从而在选择过程中保持种群的多样性,防止算法过早收敛到局部最优解,导致局部最优问题。图 4 为拥挤距离的示例图,有关公式为:

$$I_d = \sum_{j=1}^m \frac{f_j^{i+1} - f_j^{i-1}}{|f_j^{\max} - f_j^{\min}|},$$

其中: I_d 表示个体 i 的拥挤距离, m 是目标函数的数量, f_j^i 是个体 i 在第 j 个目标函数上的值, f_j^{\max} 和 f_j^{\min} 分别是第 j 个目标的最大值和最小值。

2.5 精英保留策略

将父代种群和子代种群合并,进行快速非支配排序和拥挤度计算。根据排序和拥挤度结果,保留最优的 N 个个体作为新的父代种群 P_{t+1} 。精英保留策略的过程如图 5 所示。

2.6 交叉和突变

2.6.1 交叉

OS 层和 MS 层分别使用基于工件的交叉(job-based crossover, JBX)和两点交叉(two-points crossover, TPX)^[20]。它们的操作步骤分别如下。

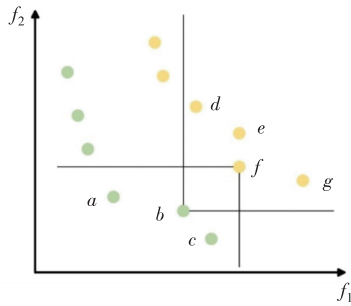


图 3 快速非支配排序示例

Fig. 3 Example for fast undominated sorting

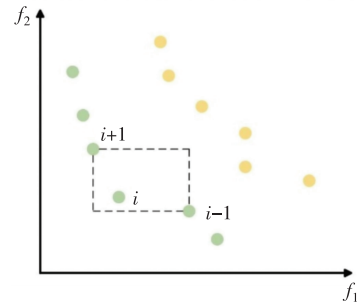


图 4 拥挤距离示例

Fig. 4 Example for crowding distance

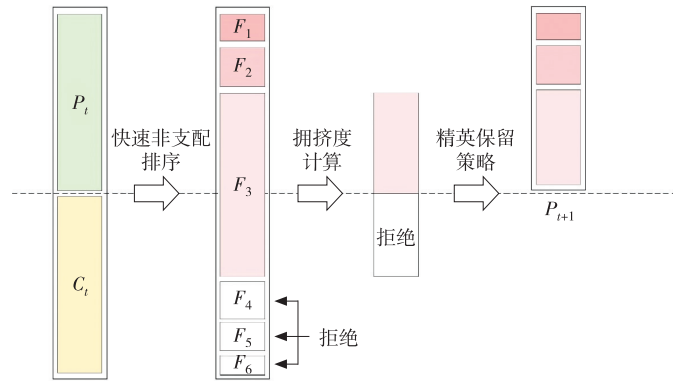


图 5 精英保留策略示例

Fig. 5 Example for elite retention strategy

JBX 操作:

第 1 步,将所有工件随机划分为 S_{j1} 和 S_{j2} 2 个子集。

第 2 步,将父代 P_1 中属于 S_{j1} 的作业复制到同一位置的子代 C_1 ;将父代 P_2 中属于 S_{j2} 的作业复制到同一位置的子代 C_2 。

第 3 步,复制 P_1 中属于 S_{j1} 的作业,然后从左到右填到 C_2 中的空位置;复制 P_2 中属于 S_{j2} 的作业,然后从左到右填到 C_1 中的空位置。

TPX 操作:

第 1 步,在 P_1 和 P_2 中随机选择 2 个位置,即 T_{P_1} 和 T_{P_2} 。

第 2 步,将 P_1 中 T_{P_1} 和 T_{P_2} 之间的基因复制到 C_2 相同位置;将 P_2 中 T_{P_1} 和 T_{P_2} 之间的基因复制到 C_1 相同位置。

第 3 步,将 P_1 的第 1 个基因和交叉起点之间的基因复制到 C_1 ;将交叉终点和 P_1 最后 1 个基因之间的基因复制到 C_2 ;将 P_2 的第 1 个基因和交叉起点之间的基因复制到相同位置的 C_2 ;将交叉终点和 P_2 最后 1 个基因之间的基因复制到相同位置的 C_1 。

JBX 和 TPX 的示例分别如图 6 和图 7 所示。

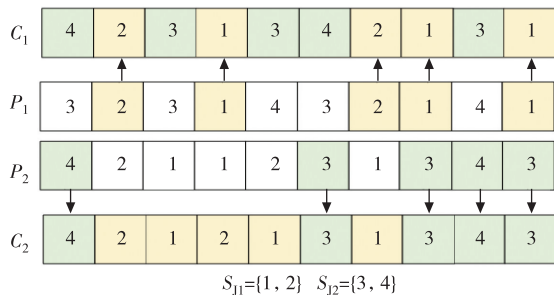


图 6 JBX 示例

Fig. 6 Example for JBX

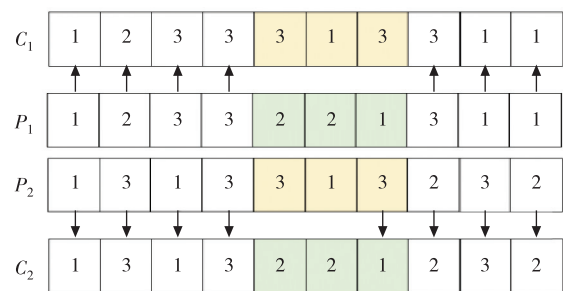


图 7 TPX 示例

Fig. 7 Example for TPX

2.6.2 变异

OS层和MS层分别使用了交换突变(swap mutation, SM)和多点突变(multi-point mutation, MPM)^[33]。它们的主要操作步骤分别如下。

SM操作:

第1步,在父代 P_1 中随机选择2个位置,即 T_{P_1} 和 T_{P_2} 。

第2步,交换 P_1 中 T_{P_1} 和 T_{P_2} 的基因生成子代 C_1 。

MPM操作:

第1步,在 P_1 中随机选择多个位置。

第2步,用可选的机器集中的其他机器随机替换这些位置上的 P_1 基因来生成 C_1 。

SM和MPM的示例分别如图8和图9所示。

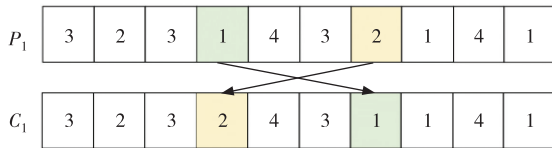


图 8 SM 示例

Fig. 8 Example for SM

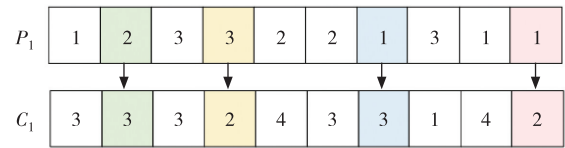


图 9 MPM 示例

Fig. 9 Example for MPM

3 数据模拟

本节首先对实验进行了设计,并对实验参数进行了调整,然后在3个不同的问题规模上用实例对模型和算法进行了检验,最后对结果进行了比较和分析。本文所有算法均在Matlab 2023b软件上编程,并在配置为2.4 GHz、8 GB RAM的Intel(R) Core(TM) i5-9300H CPU的电脑上运行。

3.1 算例生成

为证明不同问题规模下INSGA-II的有效性和优越性,本方案按工件数量分为3组,第1组有10个小规模案例(11×6),包括在6台机器上加工的11个工件。第2组由10个中规模案例(30×9)组成,包括在9台机器上加工的30个工件。第3组由10个大规模案例(50×12)组成,包括在12台机器上加工的50个工件。由于INSGA-II的随机性,每次运行的结果都可能不同。因此,每个案例都要重复运行10次以减少随机性的影响。此外,为了验证算法的适用性,所有案例都是随机生成的,数据按照以下规则生成(数据默认四舍五入到小数点后2位,功率单位为kW,能耗单位为kW·h,时间单位为min):机器生产功率从[4,10]范围内随机选择,待机功率从[0.4,3]范围内随机选择,工序加工时间从[5,10]范围内随机选择^[34-36]。所需机器根据可用机器总数随机选择^[33],每个工件包含的工序数量在[2,4]范围内选择。交叉概率为0.7,变异概率为0.3,种群大小为80,迭代次数为1000。

3.2 均衡机器能耗有效性验证

为了验证本文提出的批处理机器环境(batch machine environment, BME)对最大完工时间和总能耗的影响,并确定合理的 α 值,采取了以下步骤:第1步,为了简化计算并快速确定合理的 α 值,选择了 α 值范围内的极值和四分位数($\alpha=[0,0.25,0.5,0.75,1]$)进行下一步计算;第2步,使用INSGA-II对小规模、中等规模和大规模的所有案例按每个 α 值运行10次,并记录每次运行得到的非支配解(在此步骤中,每个情况在每个 α 值下生成10组非支配解);第3步,计算每组非支配解的各个目标的平均值;第4步,分别计算每个实例在每个 α 值下生成的10组非支配解中各个目标的平均值;最后,根据第4步的结果计算出最优值的平均值。计算结果如表5所示。为便于理解,以 $\alpha=0$ 时的小规模案例为例,绘制计算平均值和最优值的流程图如图10所示。

为了更直观地表现数据的变化趋势,根据表5的数据绘制不同 α 值下问题规模小、中、大时各目标函数值的散点图,如图11~16所示。

为了准确了解数值变化的幅度,采用一阶差分法计算了表5中各目标函数在不同规模 and 不同 α 值下的一阶差分,计算结果如表6所示。假定存在 $x_1, x_2, x_3, \dots, x_n$ 的数值序列,一阶差分计算公式为:

$$\Delta x_i = x_{i+1} - x_i。$$

表 5 目标函数的相关值($\alpha=[0, 0.25, 0.5, 0.75, 1]$)

Tab. 5 Related values of objective function ($\alpha=[0, 0.25, 0.5, 0.75, 1]$)

α	问题规模	平均值				最优值			
		f'_1	f'_2	f_1	f_2	f'_1	f'_2	f_1	f_2
0	11×6	100.56	189.81	48.46	189.81	78.76	181.79	46.01	181.79
0.25	11×6	75.36	191.12	49.09	162.18	54.62	180.65	45.88	150.50
0.5	11×6	45.56	209.37	49.57	127.47	13.91	176.56	46.14	109.07
0.75	11×6	38.25	227.27	49.75	85.50	5.71	211.31	45.71	61.76
1	11×6	36.76	231.50	49.52	36.76	4.48	190.98	45.81	4.48
0	30×9	144.65	327.68	86.51	327.68	132.39	320.55	83.92	320.55
0.25	30×9	122.59	333.79	85.70	280.99	112.05	325.91	83.92	273.49
0.5	30×9	81.36	354.80	86.21	218.08	55.52	345.80	83.74	209.08
0.75	30×9	68.13	366.56	86.72	142.74	48.90	352.29	84.00	129.93
1	30×9	61.18	377.27	87.31	61.18	41.46	362.88	84.05	41.46
0	50×12	130.49	428.25	92.65	428.25	120.86	418.45	91.36	418.45
0.25	50×12	117.92	426.35	92.96	349.24	111.69	416.88	91.25	342.07
0.5	50×12	88.04	448.85	95.41	268.45	70.12	438.54	91.31	260.74
0.75	50×12	69.28	469.43	95.58	169.32	50.34	452.35	82.53	157.28
1	50×12	63.79	488.76	95.24	63.79	46.99	420.71	91.81	46.99

注： f'_1 是最小化机器之间的能耗差异，表示的是均衡机器能耗的效果； f'_2 表示总能耗； f_1 表示的是最小化最大完工时间； f_2 表示 f'_1 与 f'_2 的加权和；下同。

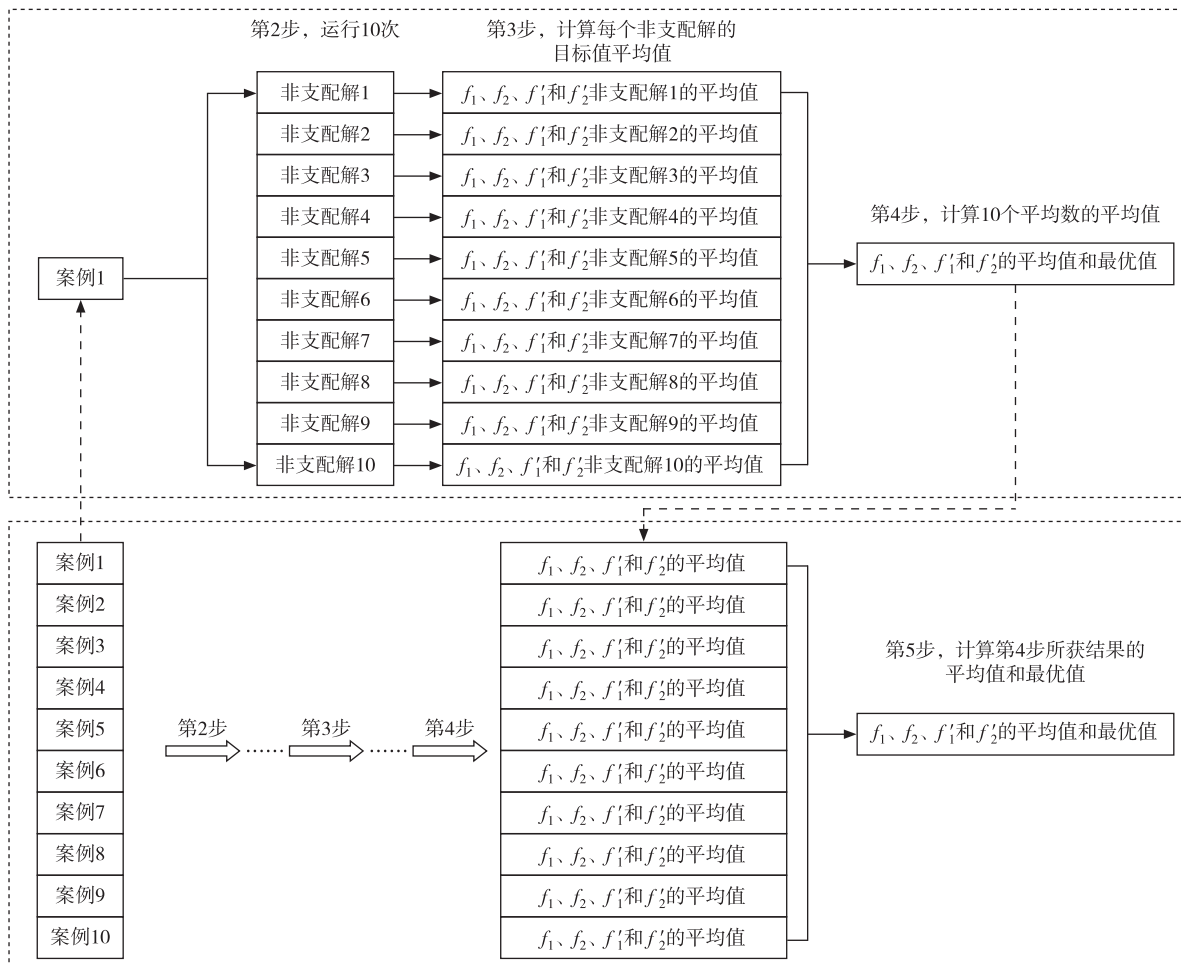


图 10 计算最优值和平均值的流程图

Fig. 10 Flowchart for calculating the optimum and average value

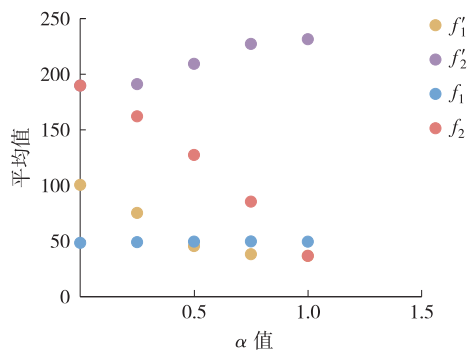


图 11 目标函数平均值(小规模)

Fig. 11 Average value of the objective function (small-scale)

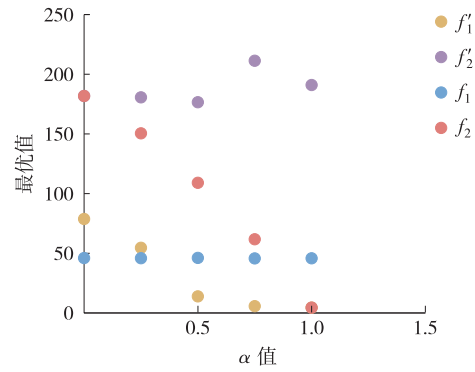


图 12 目标函数最优值(小规模)

Fig. 12 Optimal value of the objective function (small-scale)

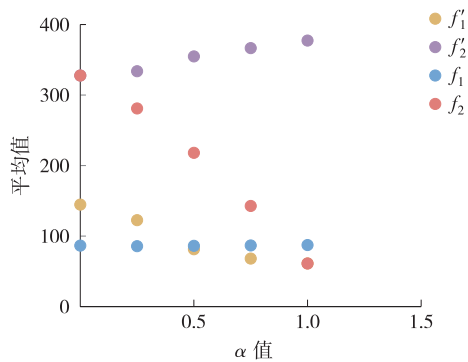


图 13 目标函数平均值(中规模)

Fig. 13 Average value of the objective function (medium-scale)

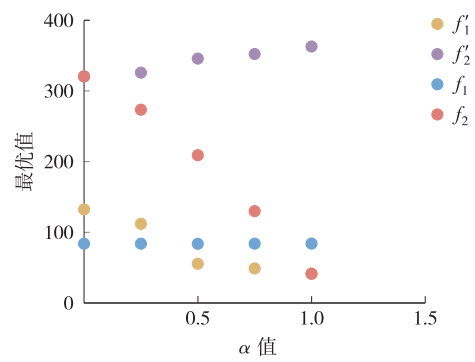


图 14 目标函数最优值(中规模)

Fig. 14 Optimal value of the objective function (medium-scale)

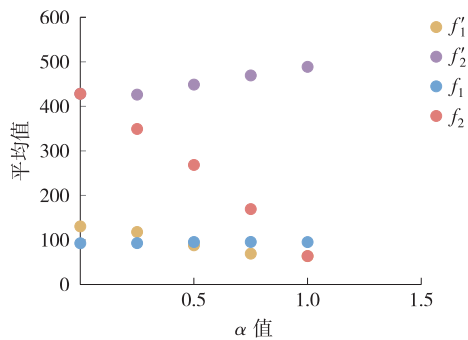


图 15 目标函数平均值(大规模)

Fig. 15 Average value of the objective function (large-scale)

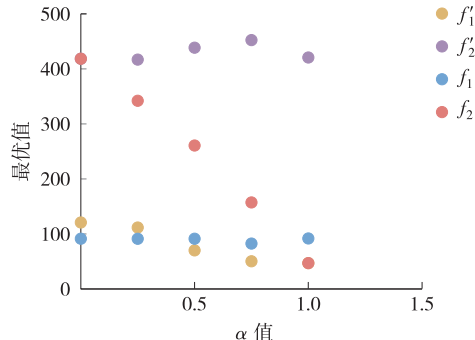


图 16 目标函数最优值(大规模)

Fig. 16 Optimal value of the objective function (large-scale)

分析表 6 和图 11~16 中的数据,可以得出以下结论:

1) BME 对最大完工时间的影响很小。根据图 11~16,在不同问题规模下随着 α 值的增大, f_1 的平均值和最优值几乎保持不变。此外如表 6 所示,当在不同数据规模下调整 α 值时, f_1 平均值的一阶差分都在 ± 1 个单位内; f_1 最优值的一阶差分除了 50×12 大规模实例的 2 个异常值(-8.78 和 9.27),其余也都在 ± 1 个单位内。这种异常值的产生原因,一方面是算法的随机性导致每次运行生成不同的初始种群,进而生成不同的最优解,当问题规模增大时初始种群的差异会被放大,使得求解结果的差异也更加明显;另一方面是当问题规模扩大至 50×12 时,实验设置的有限运行次数难以对如此庞大的解空间进行充分探索和覆盖,从而加剧了最优解的异常波动。综上所述,本文提出的 BME 对完工时间的影响很小。

2) BME 对总能耗的影响较大。根据图 11~16,在不同问题规模下随着 α 值的增加,总能耗在大多数情况下呈上升趋势,其中 $\alpha = 0.5$ 时增加最为明显,表 6 数据进一步揭示,当 $\alpha = 0.25$ 时, f_2' 在 3 个问题规模中下降趋

势均最为明显,并且 f'_1 的下降幅度在小规模和中规模案例中也仅次于 $\alpha=0.5$ 时,因此, $\alpha=0.25$ 可以在有效降低总能耗同时均衡机器能耗;当 $\alpha=0.5$ 时,虽然均衡机器能耗的效果最明显,但 f'_2 的增幅也明显较大,这表明这一选择有利于均衡机器能耗,适用于需要优化 f'_1 的情况;当 $\alpha=0.75$ 或 $\alpha=1$ 时, f'_1 的减少幅度变小同时 f'_2 的增加幅度却很大,导致总能耗的增加超过了机器间能耗差异的缩小。因此, $\alpha=0.75$ 和 $\alpha=1.0$ 并非理想的选择。综上所述, $\alpha=0.25$ 可以有效控制总能耗,同时在一定程度上减小机器能耗差异; $\alpha=0.5$ 虽然对均衡机器能耗最有效,但总能耗有明显增加; $\alpha=0.75$ 和 $\alpha=1$ 在追求机器能耗平衡时导致能耗大幅增加,因而不适合使用。

表 6 相关数值的一阶差分 ($\alpha=[0, 0.25, 0.5, 0.75, 1]$)Tab. 6 First-order difference of the related values ($\alpha=[0, 0.25, 0.5, 0.75, 1]$)

α	问题规模	平均值的一阶差分				最优值的一阶差分			
		f'_1	f'_2	f_1	f_2	f'_1	f'_2	f_1	f_2
0	11×6								
0.25	11×6	-25.20	1.31	0.63	-27.63	-24.15	-1.14	-0.13	-31.29
0.5	11×6	-29.80	18.25	0.48	-34.72	-40.71	-4.09	0.26	-41.44
0.75	11×6	-7.32	17.90	0.18	-41.97	-8.20	34.75	-0.43	-47.31
1	11×6	-1.49	4.23	-0.22	-48.74	-1.22	-20.33	0.09	-57.27
0	30×9								
0.25	30×9	-22.06	6.11	-0.81	-46.69	-20.34	5.36	0.00	-47.06
0.5	30×9	-41.23	21.00	0.51	-62.91	-56.53	19.89	-0.18	-64.41
0.75	30×9	-13.23	11.76	0.51	-75.34	-6.62	6.50	0.26	-79.15
1	30×9	-6.95	10.72	0.59	-81.56	-7.44	10.59	0.04	-88.47
0	50×12								
0.25	50×12	-12.57	-1.90	0.31	-79.00	-9.16	-1.56	-0.10	-76.38
0.5	50×12	-29.88	22.51	2.45	-80.79	-41.57	21.65	0.06	-81.33
0.75	50×12	-18.76	20.58	0.17	-99.13	-19.78	13.82	-8.78	-103.46
1	50×12	-5.49	19.32	-0.34	-105.53	-3.36	-31.64	9.27	-110.29

注:数据值为“+”表示增加,数据值为“-”表示减少,下同。

至此可以初步得出结论,将权重 α 设置在 $[0.25, 0.5]$ 的范围内能获得更好的优化结果,同时优化总能耗、机器能耗均衡和最大完工时间。为了确定一个更具体的 α 值,既能有效平衡机器能耗,又能优化总能耗,分别将 α 设为 0.25、0.3、0.35、0.4、0.45 和 0.5,再次进行计算,计算结果见表 7。

为了更直观地反映表 7 中数据的变化,根据表 7 中的数据,分别绘制了问题规模为 11×6、30×9、50×12 时不同 α 值下各目标函数值的散点图,如图 17~22 所示。此外,为了准确看出数值变化的幅度,采用一阶差分法计算了不同 α 值下小、中、大规模的各目标函数平均值和最优值的一阶差分,计算结果见表 8。

分析表 8 和图 17~22 中的数据可以得出以下结论:对于 f'_1 ,平均值和最优值的下降幅度在 $\alpha=0.35$ 时最大;对于 f'_2 ,除了中规模的异常值外,其余案例的平均值和最优值增幅在 0.25~0.5 范围内都相对稳定。这表明当 $\alpha=0.35$ 时,无论是小规模、中规模还是大规模案例, f'_1 的优化效果都比其他取值明显,并且在大多数情况下 f'_2 只出现了小幅增长。因此, $\alpha=0.35$ 时可以达到更好的协同优化效果。

3.3 INSGA-II 性能验证

本节通过大量数值实验证明了 INSGA-II 的优越性、多样性和收敛性。首先,在后续 3.3.1 小节中,将 INSGA-II 与广泛用于解决多目标优化问题多目标粒子群算法(multi-objective particle swarm optimization, MOPSO)进行了比较,通过比较 2 种算法得到的目标函数值,证明 INSGA-II 的优越性;随后,在第 3.3.2 小节中,分别比较了 INSGA-II 和 MOPSO 的超体积(hypervolume, HV)和反向世代距离(inverted generational distance, IGD),事实证明, INSGA-II 具有更好的收敛性和多样性;最后,在第 3.3.3 小节中,通过比较 INSGA-II、MOPSO 和 NSGA-II 的迭代曲线,证明 INSGA-II 与 NSGA-II 相比具有更快的求解速度。

表 7 目标函数的相关值 ($\alpha=[0.25,0.3,0.35,0.4,0.45,0.5]$)
 Tab.7 Related values of objective function ($\alpha=[0.25, 0.3, 0.35, 0.4, 0.45, 0.5]$)

α	问题规模	平均值				最优值			
		f'_1	f'_2	f_1	f_2	f'_1	f'_2	f_1	f_2
0.25	11×6	75.36	191.12	49.09	162.18	54.62	180.65	45.88	150.50
0.3	11×6	70.61	194.13	49.36	157.08	39.49	184.54	45.93	144.51
0.35	11×6	59.45	196.72	50.44	148.68	22.10	185.55	46.21	133.16
0.4	11×6	56.42	200.53	49.27	142.89	21.43	189.51	45.92	125.52
0.45	11×6	54.25	203.12	49.31	136.13	17.66	191.99	45.88	106.61
0.25	30×9	122.59	333.79	85.70	280.99	112.05	325.91	83.92	273.49
0.3	30×9	113.14	333.93	85.41	267.69	101.35	329.33	83.90	263.87
0.35	30×9	88.10	346.72	87.41	256.21	74.70	340.26	83.72	252.29
0.4	30×9	81.16	349.67	87.00	242.27	66.85	341.85	83.86	214.07
0.45	30×9	79.37	353.91	86.76	230.37	61.87	347.44	83.93	225.03
0.5	30×9	81.36	354.80	86.21	218.08	55.52	345.80	83.74	209.08
0.25	50×12	117.92	426.35	92.96	349.24	111.69	416.88	91.25	342.07
0.3	50×12	116.04	428.80	93.25	334.97	107.89	421.18	91.46	329.42
0.35	50×12	106.13	431.55	93.69	317.65	98.40	421.40	91.76	280.85
0.4	50×12	100.21	435.86	94.03	301.60	87.43	430.04	91.54	295.87
0.45	50×12	94.95	442.25	94.45	285.96	79.03	434.87	91.49	278.82
0.5	50×12	88.04	448.85	95.41	268.45	70.12	438.54	91.31	260.74
0.5	11×6	45.56	209.37	49.57	127.47	13.91	176.56	46.14	109.07

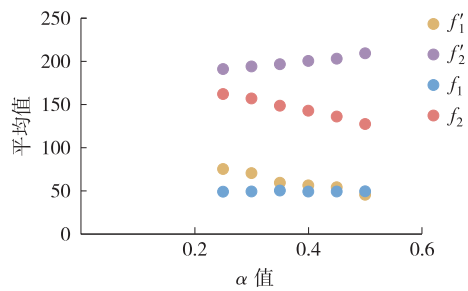


图 17 目标函数平均值(小规模)

Fig.17 Average value of the objective function (small-scale)

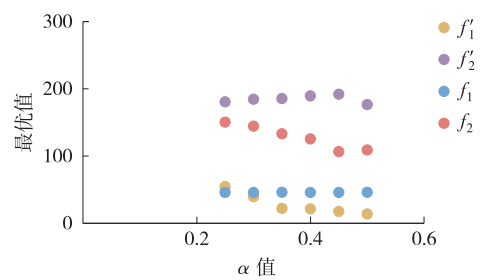


图 18 目标函数最优值(小规模)

Fig.18 Optimal value of the objective function (small-scale)

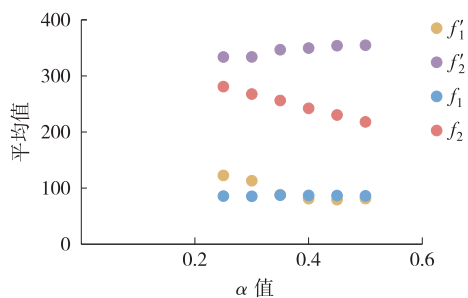


图 19 目标函数平均值(中规模)

Fig.19 Average value of the objective function (medium-scale)

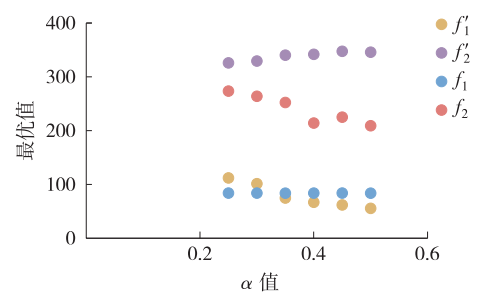


图 20 目标函数最优值(中规模)

Fig.20 Optimal value of the objective function (medium-scale)

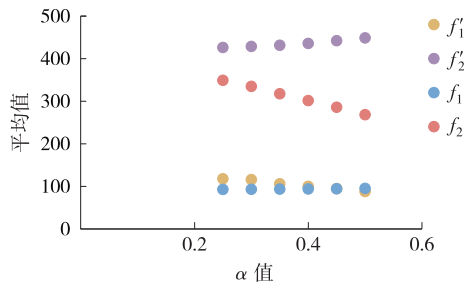


图 21 目标函数平均值(大规模)

Fig. 21 Average of the objective function (large-scale)

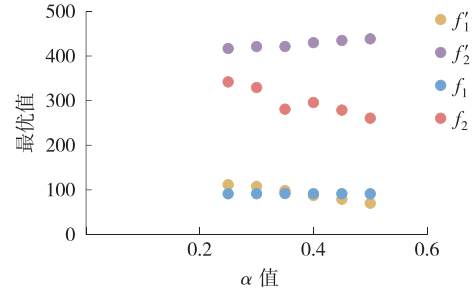


图 22 目标函数最优值(大规模)

Fig. 22 Optimal value of the objective function (large-scale)

表 8 相关数值的一阶差分($\alpha=[0.25, 0.3, 0.35, 0.4, 0.45, 0.5]$)

Tab. 8 First-order difference of the related values ($\alpha=[0.25, 0.3, 0.35, 0.4, 0.45, 0.5]$)

α	问题规模	平均值				最优值			
		f_1'	f_2'	f_1	f_2	f_1'	f_2'	f_1	f_2
0.25	11×6								
0.3	11×6	-4.75	3.01	0.26	-5.11	-15.12	3.89	0.05	-5.99
0.35	11×6	-11.16	2.59	1.08	-8.40	-17.39	1.01	0.28	-11.35
0.4	11×6	-3.02	3.81	-1.17	-5.79	-0.68	3.96	-0.29	-7.64
0.45	11×6	-2.18	2.59	0.04	-6.76	-3.77	2.48	-0.04	-18.91
0.5	11×6	-8.68	6.24	0.26	-8.66	-3.75	-15.43	0.26	2.46
0.25	30×9								
0.3	30×9	-9.45	0.14	-0.30	-13.30	-10.70	3.42	-0.02	-9.62
0.35	30×9	-25.04	<u>12.79</u>	2.00	-11.49	-26.65	<u>10.93</u>	-0.18	-11.58
0.4	30×9	-6.94	2.95	-0.41	-13.94	-7.85	1.59	0.14	-38.23
0.45	30×9	-1.79	4.24	-0.25	-11.90	-4.98	5.59	0.06	10.96
0.5	30×9	1.99	0.89	-0.54	-12.29	-6.35	-1.64	-0.19	-15.95
0.25	50×12								
0.3	50×12	-1.88	2.45	0.29	-14.27	-3.81	4.30	0.21	-12.65
0.35	50×12	-9.91	2.75	0.44	-17.32	-9.49	0.22	0.29	-48.56
0.4	50×12	-5.92	4.31	0.34	-16.05	-10.97	8.65	-0.22	15.02
0.45	50×12	-5.25	6.39	0.42	-15.63	-8.40	4.83	-0.05	-17.05
0.5	50×12	-6.91	6.61	0.96	-17.52	-8.90	3.66	-0.18	-18.08

注:下划线表示该值为异常值;加粗的数据表示最优值。

设置问题规模为 11×6、30×9 和 50×12,种群规模为 80, $\alpha=0.35$,迭代次数为 1 000。在每种情况下算法各执行 10 次。

3.3.1 INSGA-II 有效性和优越性验证

为了验证本文提出的 INSGA-II 在解决该问题中的有效性和优越性,将它与 MOPSO^[37]进行了比较。MOPSO 广泛应用于多目标优化领域,具有良好的全局搜索能力和适应性,易于实现且计算效率高。分别使用 MOPSO 和 INSGA-II 计算了每种情况下目标函数的平均值,计算过程如图 10 所示(此处仅使用 3.2 节的第 1 步至第 2 步),计算结果如表 9 所示。

根据表 9 中数据,在对比 INSGA-II 与 MOPSO 所获非支配解的目标函数值时,INSGA-II 展现出明显优势。具体而言,在小规模案例中,通过 INSGA-II 得到的 f_1 、 f_2 、 f_1' 和 f_2' 值在大多数情况下都小于通过 MOPSO 得到的值,且在中规模 and 大规模案例中,通过 INSGA-II 得到的 f_1 、 f_2 、 f_1' 和 f_2' 值也小于通过 MOPSO 得到的值。因此,本文采用的 INSGA-II 在 3 个规模案例下都能获得较好的解,尤其在解决中、大规模问题时优势更为突出。

表 9 f_1, f_2, f'_1, f'_2 的比较结果
 Tab. 9 Comparison results of f_1, f_2, f'_1 and f'_2

案例	问题规模	平均值							
		MOPSO				INSGA-II			
		f_1	f_2	f'_1	f'_2	f_1	f_2	f'_1	f'_2
1	11×6	51.15	159.00	74.47	204.51	47.79	155.28	69.95	201.23
2	11×6	51.74	155.82	58.34	208.30	48.14	148.14	65.19	192.81
3	11×6	50.70	158.39	79.57	200.84	47.78	154.30	75.96	196.49
4	11×6	50.54	160.81	82.88	202.77	49.68	156.61	76.60	199.68
5	11×6	51.62	154.73	66.62	202.17	48.02	154.05	52.93	208.50
6	11×6	50.56	162.50	81.67	206.03	47.37	154.58	82.58	193.35
7	11×6	50.88	160.33	80.76	203.17	49.68	151.06	70.21	194.59
8	11×6	50.70	156.52	64.41	206.13	47.68	153.93	69.76	199.26
9	11×6	50.94	160.66	77.36	205.52	52.32	139.42	53.92	185.46
10	11×6	50.45	160.84	85.06	201.64	48.54	155.55	58.33	207.90
11	30×9	88.87	266.90	108.35	352.28	86.50	254.36	89.74	343.00
12	30×9	89.61	264.59	108.74	348.51	87.14	255.26	98.48	339.68
13	30×9	89.50	263.94	106.41	348.77	85.28	254.53	86.67	344.92
14	30×9	87.78	270.78	113.58	355.43	84.83	254.47	96.67	339.44
15	30×9	88.39	265.68	111.91	348.48	86.15	256.85	99.85	341.39
16	30×9	88.38	267.85	113.62	350.89	87.67	256.82	93.64	344.69
17	30×9	89.81	263.09	105.99	347.69	85.46	262.87	95.82	352.82
18	30×9	89.80	260.46	95.68	349.19	86.07	255.96	86.83	347.03
19	30×9	88.23	263.57	101.25	350.97	84.94	255.50	91.74	343.67
20	30×9	90.76	260.44	96.86	348.52	86.97	253.26	97.25	337.27
21	50×12	95.48	331.51	122.85	443.86	94.79	313.58	104.51	426.16
22	50×12	95.01	329.25	120.33	441.74	92.89	312.91	103.66	425.58
23	50×12	95.90	327.79	111.39	444.31	92.84	318.41	100.77	435.60
24	50×12	95.47	331.20	116.35	446.89	94.42	315.03	116.65	421.85
25	50×12	94.50	330.22	121.37	442.68	94.88	315.60	99.13	432.17
26	50×12	95.26	323.75	98.37	445.10	92.61	317.62	114.12	427.20
27	50×12	97.15	328.73	114.81	443.93	92.99	312.41	108.59	422.16
28	50×12	94.56	335.52	118.86	452.18	92.38	316.67	102.66	431.91
29	50×12	95.28	332.26	120.97	446.03	93.12	316.07	92.69	436.35
30	50×12	95.53	326.38	107.41	444.29	94.75	315.78	108.11	427.60

注:加粗的数据是更优解。

3.3.2 INSGA-II 多样性和收敛性验证

1) HV:该指标可同时评估收敛性和多样性^[29]。HV 值越高表示算法整体性能越好,它的计算公式为:

$$V_{HV} = \delta \left(\bigcup_{i=1}^{|S|} v_i \right),$$

其中: V_{HV} 解集 S 的超体积指标; $\delta(\cdot)$ 为勒贝格测度; $|S|$ 为算法求得的非支配解集 S 中解的个数; $\bigcup_{i=1}^{|S|} v_i$ 表示所有 v_i 的并集,即被解集 S 所支配的区域,在二维目标空间中计算该并集区域的面积; $v_i = \{y \in \mathbf{R}^m \mid y \leq f(x_i) \text{ 且 } y \geq r\}$ 。

2) IGD: 该指标反映算法的收敛性和多样性^[29]。IGD 值越低表示性能越好, 计算公式为:

$$\text{IGD}(P, P^*) = \frac{\sum_{x \in P^*} \min_{y \in P} d(x, y)}{|P^*|},$$

其中: P^* 为真实帕累托前沿的非支配解集, $|P^*|$ 为 P^* 的大小, P 为算法求得的近似帕累托前沿的非支配解集, x 表示 P^* 前沿中的一个参考点, y 表示 P 前沿中的一个解, $d(x, y)$ 表示 x 与 y 之间的欧氏距离, $\text{IGD}(P, P^*)$ 越小, 说明 P 越好。

HV 和 IGD 的计算结果如表 10 所示。

表 10 MOPSO 和 INSGA-II 的 HV 和 IGD
Tab. 10 HV and IGD of MOPSO and INSGA-II

实例	问题规模	MOPSO		INSGA-II	
		HV	IGD	HV	IGD
1	11×6	0.699	0.283	0.920	0
2	11×6	0.857	0.201	0.854	0.216
3	11×6	0.798	0.265	0.806	0.034
4	11×6	0.874	0.127	0.958	0
5	11×6	0.645	0.359	0.898	0
6	11×6	0.813	0.204	0.802	0.069
7	11×6	0.799	0.200	0.884	0.026
8	11×6	0.530	0.407	0.851	0
9	11×6	0.855	0.128	0.960	0
10	11×6	0.751	0.243	0.915	0
11	30×9	0.495	0.605	0.919	0
12	30×9	0.511	0.553	0.925	0
13	30×9	0.355	0.867	0.912	0
14	30×9	0.394	0.848	0.945	0
15	30×9	0.335	0.746	0.878	0
16	30×9	0.712	0.283	0.963	0
17	30×9	0.693	0.361	0.935	0
18	30×9	0.564	0.452	0.917	0
19	30×9	0.376	0.827	0.939	0
20	30×9	0.329	0.960	0.920	0
21	50×12	0.416	0.699	0.905	0
22	50×12	0.038	1.652	0.882	0
23	50×12	0.550	0.553	0.945	0
24	50×12	0.534	0.634	0.968	0
25	50×12	0.602	0.402	0.928	0
26	50×12	0.444	0.735	0.918	0
27	50×12	0.345	0.852	0.963	0
28	50×12	0.495	0.711	0.942	0
29	50×12	0.280	0.931	0.860	0
30	50×12	0.672	0.363	0.949	0

注: 粗体字的数据是更优解。

根据表 10 中数据,可以得出以结论:1) 在大多数测试实例下,INSGA-II 的 IGD 计算结果均小于 MOPSO 的 IGD 计算结果,这说明 INSGA-II 在收敛性和多样性方面均优于 MOPSO;2) 在大多数案例中,INSGA-II 的 HV 计算结果大于 MOPSO,这与 IGD 计算结果相对应,进一步验证了 INSGA-II 在求解 GFJSP-BMEC 时能获得更好的 Pareto 前沿;3)与 MOPSO 相比,本文采用的 INSGA-II 的 IGD 和 HV 计算结果在 30 个案例下更稳定,这表明本研究采用的 INSGA-II 具有更好的鲁棒性。

3.3.3 INSGA-II 收敛性验证

由于 INSGA-II 在求解 GFJSP-BMEC 上采用了基于能耗贪婪式插入解码方法,因此,该算法能够在一定程度上加快收敛速度,为了更好地验证 INSGA-II 在求解 GFJSP-BMEC 时对收敛速度的实际影响,本节将未改进的 NSGA-II 引入作为对照,对 INSGA-II、NSGA-II 和 MOPSO 的不同规模下收敛所需的迭代次数进行了比较,并分别绘制了 3 种规模下 f_1 和 f_2 的迭代图,如图 23~28 所示。实验结果表明,INSGA-II 在收敛速度方面具有明显优势。

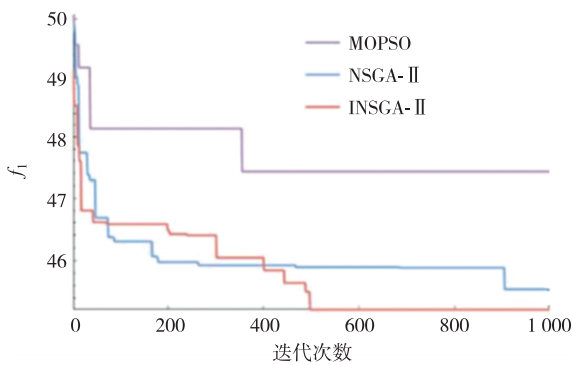


图 23 小规模 f_1 的迭代图

Fig. 23 Iteration diagram for f_1 at small-scale

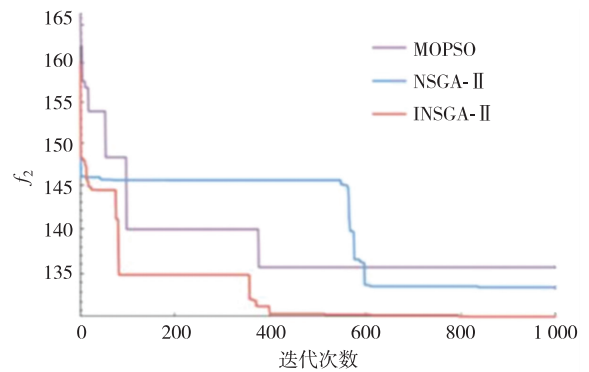


图 24 小规模 f_2 的迭代图

Fig. 24 Iteration diagram for f_2 at small-scale

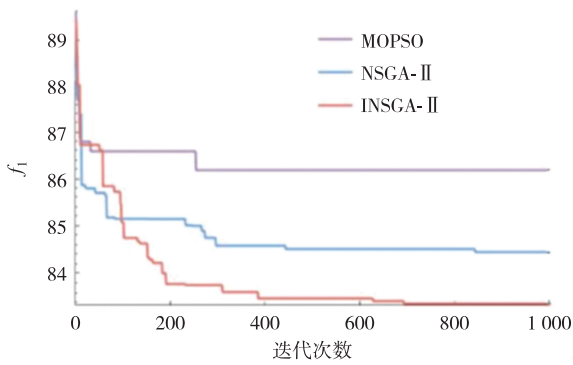


图 25 中规模 f_1 的迭代图

Fig. 25 Iteration diagram for f_1 at medium-scale

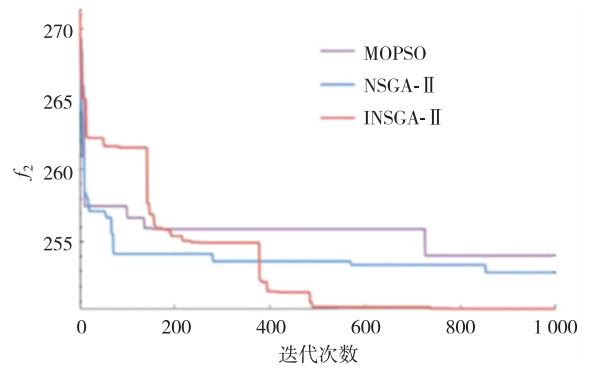


图 26 中规模 f_2 的迭代图

Fig. 26 Iteration diagram for f_2 at medium-scale

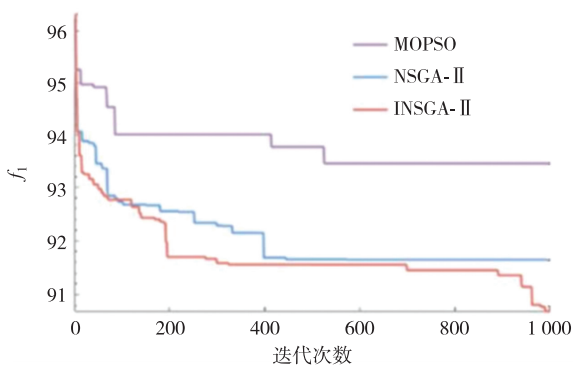


图 27 大规模 f_1 的迭代图

Fig. 27 Iteration diagram for f_1 at large-scale

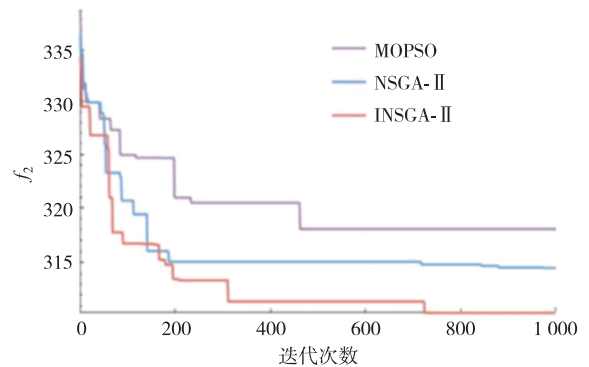


图 28 大规模 f_2 的迭代图

Fig. 28 Iteration diagram for f_2 at large-scale

从图 23~28 可以看出: INSGA-II 在小、中规模案例下能在 700 次迭代内达到收敛, 在大规模案例下亦可于 1 000 次迭代内完成收敛; 与之形成鲜明对比的是, NSGA-II 和 MOPSO 在 3 种规模下均未能在 1 000 次迭代内收敛, 收敛速度远不如 INSGA-II。这充分证明, INSGA-II 在求解 GFJSP-BMEC 时可以加快收敛速度, 在更短的时间内找到最优解。

4 结论

本文将均衡机器能耗这一优化目标引入 GFJSP 中, 旨在减小不同机器之间的能耗差异, 并防止某些机器长期在高能耗状态下运行发生故障扰乱生产计划, 而其他机器则处于闲置或低能耗状态导致机器利用率不高造成资源浪费的问题。为此, 本文建立了一个多目标优化模型来同时最小化最大完工时间以及总能耗与机器间能耗差异的加权和, 针对这个模型, 采用 INSGA-II 作为求解方法, 并开展了大量的数值实验。本文一方面通过调整权重系数 α , 分析机器能耗均衡性与总能耗之间的关联, 验证新提出的目标是否会影响最大完成时间和总能耗, 并给出 α 的合理取值; 另一方面则将 INSGA-II、MOPSO 和 NSGA-II 进行比较, 证明了 INSGA-II 在求解 GFJSP-BMEC 时的优势。在未来的研究中, 将重点优化解决这一问题的算法; 此外, 还将引入机器开关机策略和加工速度调整等操作, 以进一步降低总能耗。

参考文献:

- [1] International Energy Agency. World energy outlook 2023[EB/OL]. [2024-09-20]. <https://www.iea.org/reports/world-energy-outlook-2023>.
- [2] 刘飞, 张华, 岳红辉. 绿色制造: 现代制造业的可持续发展模式[J]. 中国机械工程, 1998, 9(6): 76-78.
LIU F, ZHANG H, YUE H H. Green manufacturing: the sustainable development model of modern manufacturing industries [J]. China Mechanical Engineering, 1998, 9(6): 76-78.
- [3] 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(3): 385-391.
WANG L, WANG J J, WU C G. Advances in green shop scheduling and optimization[J]. Control and Decision, 2018, 33(3): 385-391.
- [4] FRENCH S. Sequencing and scheduling: an introduction to the mathematics of the job-shop[M]. Chichester: Ellis Horwood, 1982.
- [5] GAREY M R, JOHNSON D S, TARJAN R E. Complexity results for multiprocessor scheduling under resource constraints[J]. SIAM Journal on Computing, 1975, 4(4): 397-411.
- [6] SUN J H, ZHANG G H, LU J, et al. A hybrid many-objective evolutionary algorithm for flexible job-shop scheduling problem with transportation and setup times[J]. Computers & Operations Research, 2021, 132(3): 105263.
- [7] ZHENG Q L, DAI W, PENG C X, et al. A novel neighborhood structure for flexible job shop scheduling problem considering quality-efficiency coupling effect[J]. Computers & Industrial Engineering, 2025, 199(1): 110735.
- [8] BRUCKER P, SCHLIE R. Job-shop scheduling with multi-purpose machines[J]. Computing, 1990, 45(4): 369-375.
- [9] 潘全科, 孙志峻, 朱剑英. 基于遗传算法的作业车间调度优化[J]. 信息与控制, 2002, 31(3): 216-218.
PAN Q K, SUN Z J, ZHU J Y. An intelligent scheduling optimization of job shop with alternative machines[J]. Information and Control, 2002, 31(3): 216-218.
- [10] YAZDANI M, AMIRI M, ZANDIEH M. Flexible job-shop scheduling with parallel variable neighborhood search algorithm[J]. Expert Systems with Applications, 2010, 37(1): 678-687.
- [11] YUAN Y, XU H. Flexible job shop scheduling using hybrid differential evolution algorithms[J]. Computers & Industrial Engineering, 2013, 65(2): 246-260.
- [12] ZENG C K, TANG J F, FAN Z P, et al. Auction-based approach for a flexible job-shop scheduling problem with multiple process plans[J]. Engineering Optimization, 2019, 51(11): 1902-1919.
- [13] 鲍蕃, 张洪亮, 丁仁曼, 等. 改进混合蛙跳算法求解考虑运输时间的柔性作业车间调度问题[J]. 重庆师范大学学报(自然科学版), 2022, 39(1): 72-78.
BAO Q, ZHANG H L, DING R M, et al. Improved shuffled frog leaping algorithm for the flexible job shop scheduling problem with transportation time[J]. Journal of Chongqing Normal University (Natural Science), 2022, 39(1): 72-78.
- [14] ZHANG Y P, YAO X, XU S. Chaos-based improved marine predators algorithm for flexible job-shop scheduling problem[J]. Journal of Mechanical Science and Technology, 2024, 38(10): 5581-5594.
- [15] SHEN L J, DAUZÈRE-PÉRÈS S, MAECKER S. Energy cost efficient scheduling in flexible job-shop manufacturing systems

- [J]. *European Journal of Operational Research*, 2023, 310(3): 992-1016.
- [16] LOU H Y, WANG X P, DONG Z M, et al. Memetic algorithm based on learning and decomposition for multiobjective flexible job shop scheduling considering human factors[J]. *Swarm and Evolutionary Computation*, 2022, 75(5): 101204.
- [17] VITAL-SOTO A, BAKI M F, AZAB A. A multi-objective mathematical model and evolutionary algorithm for the dual-resource flexible job-shop scheduling problem with sequencing flexibility[J]. *Flexible Services and Manufacturing Journal*, 2023, 35(3): 626-668.
- [18] SUN J H, ZHANG Z, ZHANG G H, et al. Multi-objective evolutionary algorithm based flexible assembly job-shop rescheduling with component sharing for order insertion[J]. *Computers & Operations Research*, 2024, 169(9): 106744.
- [19] WEI G Y, YE C M, XU J N. Shared manufacturing-based distributed flexible job shop scheduling with supply-demand matching [J]. *Computers & Industrial Engineering*, 2024, 189(3): 109950.
- [20] 张洪亮, 徐公杰, 鲍蓓, 等. 考虑运输时间的分布式柔性作业车间绿色调度[J]. *中国机械工程*, 2022, 33(21): 2554-2563.
ZHANG H L, XU G J, BAO Q, et al. Distributed flexible job shop green scheduling with transportation time[J]. *China Mechanical Engineering*, 2022, 33(21): 2554-2563.
- [21] CHEN X L, LI J Q, XU Y. Q-learning based multi-objective immune algorithm for fuzzy flexible job shop scheduling problem considering dynamic disruptions[J]. *Swarm and Evolutionary Computation*, 2023, 83(8): 101414.
- [22] ZHANG W K, ZHENG Y F, AHMAD R. An energy-efficient multi-objective integrated process planning and scheduling for a flexible job-shop-type remanufacturing system[J]. *Advanced Engineering Informatics*, 2023, 56(2): 102010.
- [23] XIAO B, ZHAO Z C, WU Y C, et al. An improved MOEA/D for multi-objective flexible job shop scheduling by considering efficiency and cost[J]. *Computers & Operations Research*, 2024, 167(7): 106674.
- [24] 蒋权, 魏静萱. 用于动态柔性作业车间调度的实时调度方法[J]. *系统仿真学报*, 2024, 36(7): 1609-1620.
JIANG Q, WEI J X. Real-time scheduling method for dynamic flexible job shop scheduling[J]. *Journal of Simulation*, 2024, 36(7): 1609-1620.
- [25] 江雨燕, 李文凯, 李艳, 等. 考虑恶化效应的多目标柔性作业车间绿色调度[J]. *系统工程*, 2024, 42(4): 28-39.
JIANG Y Y, LI W K, LI Y, et al. Green scheduling of multi-objective flexible job shop considering deterioration effect[J]. *Systems Engineering*, 2024, 42(4): 28-39.
- [26] ZHANG H L, QIN C Q, XU G J, et al. An energy-saving distributed flexible job shop scheduling with machine breakdown[J]. *Applied Soft Computing*, 2024, 167: 112276.
- [27] LIU Q, TIAN Y Q, WANG C, et al. Flexible job-shop scheduling for reduced manufacturing carbon footprint[J]. *Journal of Manufacturing Science and Engineering*, 2018, 140(6): 061013.
- [28] 李益兵, 黄炜星, 吴锐. 基于改进人工蜂群算法的多目标绿色柔性作业车间调度研究[J]. *中国机械工程*, 2020, 31(11): 1344-1350.
LI Y B, HUANG W X, WU R. Research on multi-objective green flexible job-shop scheduling based on improved ABC algorithm[J]. *China Mechanical Engineering*, 2020, 31(11): 1344-1350.
- [29] WANG Z X, HE M W, WU J, et al. An improved MOEA/D for low-carbon many-objective flexible job shop scheduling problem[J]. *Computers & Industrial Engineering*, 2024, 188(2): 109926.
- [30] QU H Q, TONG X M, CAI M L, et al. Energy-saving scheduling strategy for variable-speed flexible job-shop problem considering operation-dependent energy consumption[J]. *Expert Systems with Applications*, 2024, 256(22): 124952.
- [31] SATHYASOFIA A, GANESHKUMAR P. Multi-objective task scheduling to minimize energy consumption and makespan of cloud computing using NSGA-II [J]. *Journal of Network and Systems Management*, 2018, 26(2): 463-485.
- [32] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [33] GONG G L, CHIONG R, DENG Q W, et al. A two-stage memetic algorithm for energy-efficient flexible job shop scheduling by means of decreasing the total number of machine restarts[J]. *Swarm and Evolutionary Computation*, 2022, 75(8): 101131.
- [34] 吕海利, 孙佳祺, 吴姝. 考虑机器能耗的绿色作业车间调度问题[J]. *运筹与管理*, 2021, 30(12): 220-225.
LÜ H L, SUN J Q, WU S. Green job shop scheduling problem considering machine energy consumption[J]. *Operations Research and Management Science*, 2021, 30(12): 220-225.
- [35] 孟磊磊, 张超勇, 肖华军, 等. 面向加工时间可控的柔性作业车间节能调度问题建模[J]. *计算机集成制造系统*, 2019, 25(5): 1062-1074.
MENG L L, ZHANG C Y, XIAO H J, et al. Mathematical modeling of energy-efficient flexible job shop scheduling problem with controllable processing times[J]. *Computer Integrated Manufacturing Systems*, 2019, 25(5): 1062-1074.
- [36] 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.

[37] COELLO C A,LECHUGA M S. MOPSO:A proposal for multiple objective particle swarm optimization[C]//Proceedings of the 2002 Congress on Evolutionary Computation, Honolulu:IEEE,2002.

Operations Research and Cybernetics

Multi-Objective Scheduling of Green Flexible Job Shop Based on Balanced Machine Energy Consumption

PU Hongyu¹, MA Ran¹, ZHANG Yuzhong²

(1. School of Management Engineering, Qingdao University of Technology, Qingdao Shandong 266520;

2. Institute of Operations Research, Qufu Normal University, Rizhao Shandong 273165, China)

Abstract: Machines are an integral part of the actual workshop production system. If machinery operates for prolonged periods under high energy consumption, it will accelerate equipment aging and increase the likelihood of malfunctions, thereby disrupting production cycles. Conversely, sustained idle states or low energy utilization can lead to significant resource waste. Therefore, achieving energy consumption equilibrium is a critical and highly valuable research area that merits systematic exploration. It addresses the green flexible job shop scheduling problem with balanced machine energy consumption (GFJSP-BMEC) and proposes an optimization model aimed at minimizing two objectives: makespan and the weighted sum of inter-machine energy consumption differences and total energy consumption. To this end, an improved non-dominated sorting genetic algorithm II (INSGA-II) is developed to optimize these objectives simultaneously. Extensive numerical experiments are conducted to verify the impact of the machine energy consumption balancing strategy on makespan and total energy consumption. Furthermore, INSGA-II is compared with non-dominated sorting genetic algorithm II and multi-objective particle swarm optimization to demonstrate its effectiveness and superiority in solving GFJSP-BMEC.

Keywords: green scheduling; multi-objective optimization; flexible job shop; balanced energy consumption; INSGA-II

(责任编辑 陈新颖)