



不确定工时下多目标柔性作业车间调度问题的进化求解方法

钟小玉¹, 韩玉艳², 姚香娟^{1,4*}, 巩敦卫^{3*}, 孙永征¹

1. 中国矿业大学数学学院, 徐州 221116

2. 聊城大学计算机学院, 聊城 252000

3. 中国矿业大学信息与控制工程学院, 徐州 221116

4. 南京大学计算机软件新技术国家重点实验室, 南京 210023

* 通信作者. E-mail: yaoxj@cumt.edu.cn, dwgong@vip.163.com

收稿日期: 2022-05-20; 修回日期: 2022-07-15; 接受日期: 2022-07-21; 网络出版日期: 2023-04-12

国家自然科学基金重点项目 (批准号: 62133015)、国家自然科学基金 (批准号: 61803192) 和中央高校基本科研业务费专项资金 (批准号: 2020ZDPMYS40) 资助

摘要 在实际工业生产中, 调度环境的复杂性与不确定性使得调度问题求解难度大大提高. 针对加工时间不确定的柔性作业车间调度问题, 采用不确定参数描述随机工时波动程度和约束条件允许违背程度, 构建工时波动服从指数分布的多目标柔性车间调度模型. 基于机会约束规划理论, 将不确定调度问题转化为加工时间确定的柔性作业车间调度问题, 求解得到一定程度上具有鲁棒性能的调度方案. 在执行过程中, 采用工序移动调整和重调度方法对作业排产方案进行动态调整. 基于双链式编码以及贪婪插入法解码规则, 提出了基于变邻域搜索的混合 NSGA-II 算法. 针对车间调度问题的多约束性和计算复杂度高等特点, 设计了基于机器选择的复合启发式规则, 包括依据概率的最小累计机器负载和最短工序加工时间规则, 以获取更加接近 Pareto 前沿的均匀分布初始种群. 采用改进工序和设备交叉策略以提高算法的全局搜索能力. 此外, 基于关键工序和机器选择的多种邻域结构, 设计了变邻域搜索策略, 以进一步提高算法的局部搜索能力. 通过 Kacem 和 Brandimarte 标准算例的数值仿真以及与多种代表算法的统计比较, 验证了所提算法的有效性. 本文所提算法为不确定柔性作业车间调度问题提供了更优的调度方案.

关键词 多目标优化, 柔性作业车间调度, NSGA-II, 变邻域搜索, 不确定工时, 机会约束规划

1 引言

制造业水平的高低直接反映一个国家生产能力的强弱, 而车间调度是制造业的关键环节. 车间调度问题的研究目的, 是在满足调度目标的前提下, 合理安排各种资源、优化生产计划, 从而提高生产效

引用格式: 钟小玉, 韩玉艳, 姚香娟, 等. 不确定工时下多目标柔性作业车间调度问题的进化求解方法. 中国科学: 信息科学, 2023, 53: 737-757, doi: 10.1360/SSI-2022-0017
Zhong X Y, Han Y Y, Yao X J, et al. An evolutionary algorithm for the multi-objective flexible job shop scheduling problem with uncertain processing time (in Chinese). Sci Sin Inform, 2023, 53: 737-757, doi: 10.1360/SSI-2022-0017

率、节约生产成本. 柔性作业车间调度问题 (flexible job-shop scheduling problem, FJSP) 是一类比传统作业车间调度更为复杂的组合优化问题. 传统作业车间调度与 FJSP 的区别在于, 前者每道工序的加工机器以及加工时间都已经确定, 而后者每道工序都有若干可选加工机器, 并且不同机器的加工时间是不一样的, 这就增加了调度的复杂性.

生产调度问题的性能指标包括时间指标 (如最大完工时间、交货期、延迟或者拖期等), 成本指标 (如生产成本等), 其他指标 (如最大机器负荷、机器总负荷、库存等). 针对单目标 FJSP, 很多专家学者提出多种智能优化算法进行求解, 取得了丰硕的研究成果. 为了最小化总能源消耗, Rakovitis 等^[1]提出一种基于分组的分解算法, 将整个问题分解为多个子问题, 有效解决大规模 FJSP. 而实际生产过程中, 通常既要求缩短周期、降低成本, 又要求设备利用率高、负载均衡. 因此, 研究多目标 FJSP 更具有现实意义. 为同时最小化完工时间和总能源消耗, Zhang 等^[2]提出一种有效的新型启发式方法 (NHM) 求解具有序列相关设置和运输时间的节能 FJSP, 并设计了 3 种策略——种群初始化、贪婪迭代解码和局部强化, 增强了 NHM 解的收敛性和多样性. 李等^[3]考虑运输资源、工件间准备时间等约束条件, 提出一种改进人工蜂群优化算法, 设计了 5 种调度邻域结构和一种机床分配邻域结构, 嵌入了模拟退火接受准则, 提升了算法的局部搜索能力和全局搜索能力. 为同时最小化最大完工时间和工件总转移时间以及同时优化最大完成时间、总延迟时间、总能耗和机器总负荷, 郑等^[4]提出一种新型人工蜂群算法, 通过两类蜜蜂彼此各异的搜索方式来避免算法陷入局部最优. 同时, 该算法将跟随对象限定为质量较好的部分引领蜂和外部档案成员, 并给出了侦查蜂 (scout) 新的处理策略, 有效解决高维多目标 FJSP. Lei 等^[5]考虑工作负荷平衡和总能源消耗的 FJSP, 提出一种基于三字符编码方法的混合蛙跳算法 (SFLA), 基于锦标赛选择机制, 利用种群和非支配集构建元丛, 每个元丛中的搜索过程在其非支配解上完成. Gao 等^[6]研究新工件插入的柔性作业车间重调度问题 (FJRP), 为衡量对现有计划所做更改量以及重调度方案质量, 将不稳定性纳入优化目标, 改进离散化的元启发式算法 DJaya 以求解 FJRP.

上述研究的多为确定环境下单目标和多目标调度问题, 然而, 实际生产车间中还存在诸多不确定因素, 比如, 加工时间波动、工艺损伤、机械故障、人工操作失误等, 致使现场作业与生产计划之间存在偏差, 表现为工件的开始加工时间与结束加工时间呈现随机性和不确定性. 本文不考虑如工件/配件随机到达、原材料延期到达等不确定因素对开始加工时间带来的随机性影响, 只考虑生产过程中的不确定因素对加工时间产生的影响. 利用插入式贪婪算法求得最佳可行开始时刻, 即开始加工时间只由工件上一道工序完工时间和被分配机器的空闲时间决定. 工序结束加工时间由工序的开始加工时间和加工时间决定, 故对开始加工时间和结束加工时间的随机性分析可转化为对加工时间的随机性分析. 例如, Goren 等^[7]采用主动调度方法研究机器故障与加工时间的不确定性, 并将机器故障这一不确定因素转化为对工序加工时间的影响; Yang 等^[8]提出, 由操作工人技术水平参差不齐、工件返工与返修、切削参数调整、刀具磨损或者更换等多种不确定因素引起的随机波动都可以直接或间接反映到工序加工时间的变化. 因此, 本文只考虑工件生产过程中遇到的不确定因素对加工时间产生的影响, 将工序加工时间作为随机变量进行分析.

随机优化方法是使用概率模型预测各种不确定因素和风险因素的一种定量分析方法, 它使得车间调度问题中的不确定性明晰化且能够估计出最终解变化的范围. 典型的随机优化模型是将系统的期望作为优化指标. Sun 等^[9]针对加工时间分别服从正态分布、高斯分布和指数分布的随机柔性调度问题, 提出一种基于马尔可夫随机场 (Markov random field, MRF) 分解策略的混合协同演化算法 (hCEA-MRF), 来最小化完工时间的期望和方差. Horng 等^[10]针对加工时间服从截断正态分布、均匀分布和指数分布 3 种情形, 提出一种两阶段仿生优化方法优化最大完工时间. Tang 等^[11]假定加工时

间的扰动服从均匀分布,引入 2 个不确定参数描述随机变量波动程度和约束条件允许违背程度,结合遗传进化的全局优化和邻域搜索的空间拓展能力提出一种鲁棒调度算法以优化最大完工时间。

上述研究主要针对最大完工时间的车间调度问题,给出了不确定加工时间的描述和度量方法,而已有文献较少考虑不确定加工时间对多目标 FJSP 的影响。此外,关于调度方案与约束条件允许违背程度的折中以及根据不确定变量的波动程度来给出调度方案的研究较少。文献 [12,13] 通过仿真实验,验证了作业时间服从渐进指数分布。鉴于此,本文从生产需求出发,研究加工时间服从指数分布的多目标 FJSP。

柔性作业车间调度算法可分为精确算法、启发式算法、神经网络算法,以及智能优化方法四类。由于时间限制,难以应用精确算法求解大规模调度问题。启发式算法过程易于实现,但往往无法获取高质量解。近年来,随着深度学习神经网络的迅速发展,深度学习神经网络在求解智能调度问题中扮演越来越重要的角色。陈等^[14]以平均延期惩罚、能耗、偏差度为目标搭建基于动态事件扰动和周期性重调度的学习环境,然后利用深度 Q 学习神经网络算法建立环境-行为评价神经网络模型作为优化模型的适应度函数,最后采用改进的量子遗传算法求解动态调度优化模型;任剑锋等^[15]提出了一种数据驱动的作业车间调度算法,以 LSTM 模型为主框架,在模型中嵌入指针网络,令当前序列中概率最大的工件优先进入缓冲区,提高了神经网络的训练速度和质量,采用训练后的模型对新问题进行求解,为求解作业车间调度问题提供了新思路;李宝帅等^[16]针对传统车间调度方法实时响应能力有限,提出一种基于深度 Q 网络的深度强化学习算法,该方法结合深度神经网络的学习能力与强化学习的决策能力,将车间调度问题视作序列决策问题,用深度神经网络拟合价值函数,使用多个调度规则作为动作空间,并设置基于机器利用率的奖励函数,不断与环境交互,从而获得每个决策点的最佳调度规则。

智能优化算法兼顾解的质量和运行速度,具有很好的求解效果。智能优化方法可分为进化算法(EA)、群智能算法(SI)和局部搜索算法(LS)等。超过一半的 FJSP 算法属于 EA^[17]。遗传算法(GA)是一种典型的进化算法,被广泛研究,但传统的遗传算法局部搜索能力差,容易陷入局部最优。NSGA-II 是 Deb 等^[18]在 2002 年提出的一种进化算法,因其在搜索 Pareto 前沿解的迅速性和维持解的多样性等方面的优势,一经提出就引起了广泛关注,并成功应用于解决各种优化问题。王亚昆等^[19]应用 NSGA-II 来解决车间运输约束导致的调度方案与实际加工偏差问题;Seng 等^[20]应用 NSGA-II 来解决低碳 FJSP,在保证生产效率的同时通过自动化管理降低生产线的空载和总能耗;Tan 等^[21]应用 NSGA-II 来解决机器受限和工人具有疲劳意识的双资源约束 FJSP。

研究表明,针对单一算法的不足,混合算法将两种及以上算法的特性进行互补,相比单一算法具有更优的适应性和鲁棒性,是研究和解决多目标 FJSP 的重要途径,尤其是混合智能优化算法与邻域搜索策略。例如,Gao 等^[22]混合离散和声算法与邻域搜索策略,基于关键块理论,针对不同的目标运用不同邻域结构,实验结果表明,该混合算法性能显著优于单一算法。Gong 等^[23]混合模因算法和邻域搜索,设计 N1, N2 两种工序邻域结构有效提升算法求解能力。赵诗奎^[24]混合遗传算法与邻域搜索,将邻域结构分两级进行,对已有邻域结构进行了无效移动的精简和有效移动的拓展,并给出近似评价方法。尽管混合邻域搜索算法目前已取得大量研究成果,然而在进行邻域搜索时仍然存在盲目、随机、过多无效移动的问题,且往往仅能优化单一目标。

本文在 NSGA-II 研究的基础上,引入变邻域搜索算法(variable neighborhood search, VNS),提出一种混合多目标遗传算法(VNS-NSGA-II)。并通过实验证明所提算法的可行性、有效性和实用性。本文具体贡献如下。

(1) 建模建立方面的创新。本文考虑了最大完工时间、最大机器负荷和机器总负荷 3 个优化目标,建立了加工时间波动服从指数分布的多目标 FJSP 模型:采用不确定变量独立变动程度以及不确定变

量波动程度来描述工件实际加工时间和理论加工时间之间的关系; 基于机会约束规划理论, 利用给定指数分布和约束条件允许违背程度下的分位点, 将不确定模型转化为确定性模型; 通过求解对等确定性模型的最优解作为预调度方案, 并在实际生产中采用工序移动调整和重调度方法对作业排产方案进行动态调整以实现计划调度方案与执行现场的同步。

(2) 算法设计方面的创新. 针对上述 3 目标优化问题, 本文提出改进的 NSGA-II 算法: 设计了一种新的种群初始化策略, 为产生分布更加均匀且更加接近 Pareto 前沿的初始种群, 针对机器选择链, 以一定的概率应用机器累计负载最小、工序加工时间最短等规则进行机器选择, 使得生成的解具有更好的性能; 给出一种基于问题特性的邻域搜索策略 (VNS), 使得邻域搜索更加精准有效的同时适用于多目标优化, 研究基于关键工序和机器选择的多种邻域结构设计变邻域搜索策略. 最后的实验结果表明, 本文算法可以降低传统 NSGA-II 在初始化、交叉、变异等方面的盲目性, 提高算法的局部搜索能力, 从而加快了算法的收敛速度。

2 不确定工时下多目标柔性作业车间调度问题

2.1 问题描述

具有不确定加工时间的 FJSP 问题可被描述为: 有 n 个待加工工件 J_i ($i = 1, 2, \dots, n$) 组成了总工件集合 \mathcal{J} , m 台加工机器 M_k ($k = 1, 2, \dots, m$) 组成总机器集合. 假设工件 J_i 包含 n_i ($n_i \geq 0$) 道工序, O_{ij} 表示工件 i 的第 j 道工序. 工序 O_{ij} 对应的可选加工机器组成的集合为 M_{ij} , 当工序 O_{ij} 在可选加工机器集 M_{ij} 上确定了加工机器后, 其相应的名义加工时间也就确定了, 记为 p_{ijk} . 由于实际生产中存在各种不确定因素, 工件加工时间会受到干扰, 在名义加工时间附近波动, 工件实际加工时间记为 \tilde{p}_{ijk} . 调度的任务是在不确定环境下确定工序的最优加工顺序以及工序的最优机器分配, 使得满足约束条件的同时系统性能指标得到优化。

针对该问题一般作以下假设: (1) 一台机器一次只能加工一个工件; (2) 一个工件一次只能被一台机器加工; (3) 不同的工件是相互独立的, 即机器在选择要处理的工件时呈现随机性, 所有工件具有相同的优先级, 并且没有顺序限制; (4) 工件一旦开始加工, 就不能中途停止; (5) 工件在不同机器上的转移时间为零; (6) 所有工件在零时刻都可以被加工。

2.2 多目标柔性作业车间调度数学模型

为了衡量计划的进度情况, 车间调度常使用最大完成时间作为最基本指标. 最大完成时间是生产效率的重要体现, 它指的是最后一道工序的完工时间. 机器负荷也是制造企业关注的一项重要指标, 测量方法包括: 最大机器负荷、总机器负荷等. 最大机器负荷是指所有机器中最大的负荷, 总机器负荷是指所有机器上的总处理时间. 最大机器负荷和总机器负荷越小, 机器分布越好. 因此, 本文采用最大完工时间、最大机器负荷和总机器负荷 3 个指标对调度方案进行评价。

结合调度甘特示意图 (如图 1 所示), 给出模型涉及的符号: T_o 表示总工序数; 一道工序在机器上的加工记为一个事件点, 机器 M_k 上的事件点 t 表示机器 M_k 上的第 t 道加工工序, 机器 M_k 包含 t_k 个事件点 ($t_k \leq T_o$); 过程 MAX 为一个足够大的正实数; PS_{kt} 表示机器 M_k 在事件点 t 的计划开始加工时间; S_{kt} 表示机器 M_k 在事件点 t 的实际开始加工时间; PE_{kt} 表示机器 M_k 在事件点 t 的计划结束加工时间; E_{kt} 表示机器 M_k 在事件点 t 的实际结束加工时间; ST_{ij} 表示工序 O_{ij} 的实际开始加工时间; ET_{ij} 表示工序 O_{ij} 的实际结束加工时间; 若工序 O_{ij} 分配给机器 M_k 的第 t 个事件点则 $Z_{ijk} = 1$, 否则 $Z_{ijk} = 0$; C_{\max} 表示所有工件最大完工时间。

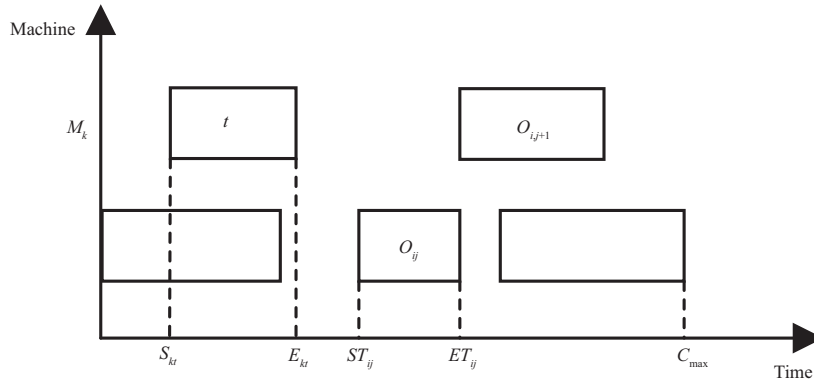


图 1 甘特示意图
Figure 1 Gantt diagram

由于最大完工时间、最大机器负荷和总机器负荷均为随机变量, 因此, 我们将其数学期望作为目标函数. 随机工时下的多目标 FJSP 数学模型为

$$\min f_1 = E[C_{\max}], \tag{1}$$

$$\min f_2 = E \left[\max_{1 \leq k \leq m} \sum_{i=1}^n \sum_{j=1}^{n_i} \sum_{t=1}^{t_k} (\tilde{p}_{ijk} \times Z_{ijkt}) \right], \tag{2}$$

$$\min f_3 = E \left[\sum_{i=1}^n \sum_{j=1}^{n_i} \sum_{k=1}^m \sum_{t=1}^{t_k} (\tilde{p}_{ijk} \times Z_{ijkt}) \right], \tag{3}$$

s.t.

$$\sum_{k \in M_{ij}} \sum_t Z_{ijkt} = 1, \quad \forall i, j, \tag{4}$$

$$\sum_i \sum_j Z_{ijkt} \leq 1, \quad \forall k, t, \tag{5}$$

$$\sum_i \sum_j Z_{ijkt} \geq \sum_i \sum_j Z_{ijk,t+1}, \quad \forall k, t < t_k, \tag{6}$$

$$E_{kt} \leq S_{k,t+1}, \quad \forall k, t < t_k, \tag{7}$$

$$E_{kt} = S_{kt} + \sum_i \sum_j (\tilde{p}_{ijk} \times Z_{ijkt}), \quad \forall k, t, \tag{8}$$

$$ST_{ij} \geq ET_{i,j-1}, \quad \forall i, j > 1, \tag{9}$$

$$ET_{ij} \leq E_{kt} + \text{MAX} \times (1 - Z_{ijkt}), \quad \forall i, j, k \in M_{ij}, t, \tag{10}$$

$$ET_{ij} \geq E_{kt} - \text{MAX} \times (1 - Z_{ijkt}), \quad \forall i, j, k \in M_{ij}, t, \tag{11}$$

$$C_{\max} \geq ET_{ij}, \quad \forall i, j. \tag{12}$$

式 (1)~(3) 给出了模型的 3 个优化目标, 即最小化最大完工时间、最大机器负荷和总机器负荷的数学期望; 式 (4) 表示一道工序只能同时由一台机器来处理; 式 (5) 表示一台机器一次最多只能加工一道工序; 式 (6) 表示各机器上, 需按照从前向后的顺序对工序进行排序; 式 (7) 表示各机器上, 一个

任务执行完毕之后才能开始新的任务; 式 (8) 表示各机器上, 一个任务一旦开始后不允许中断; 式 (9) 表示对于任意工件, 各工序完成之后才能开始下一道工序; 式 (10) 和 (11) 表示各机器事件点对应各工序事件点; 式 (12) 表示最大完工时间为所有工件的完工时间中的最大值.

2.3 基于机会约束规划的模型确定性转换

将带有随机性目标函数的数学期望作为目标函数是一种规避中性风险的鲁棒调度方法, 目的是得到潜在加工时间随机扰动下平均性能优异的调度方案. 采用 Monte Carlo 随机仿真方法根据工序加工时间给出的概率分布随机采样, 产生一组实际加工中可能出现的结果称为一次抽样. 由大数定律可知当随机模拟次数 $L \rightarrow \infty$ 时, 可采取 L 次模拟的均值作为目标函数数学期望的估计, 具体 L 值可根据对优化目标评价精度的要求设定.

对于含有随机变量的约束, 本文基于机会约束规划将不确定约束转化为确定约束. 机会约束规划由 Charnes 和 Cooper^[25] 提出, 主要针对约束条件中存在随机变量, 且必须在观测到随机变量实现之前作出决策的情况. 考虑到所做决策在不利情况发生时可能不满足约束条件, 而采取一种原则, 即允许所做决策在一定程度上不满足约束条件, 但该决策应使约束条件满足的概率不小于某一置信水平. 并根据事先给定的置信水平, 将机会约束转化为其确定等价类. 为了将机会约束规划技术应用在模型中, 将式 (8) 改写为如下不等式约束:

$$(S_{kt} - E_{kt}) + \sum_i \sum_j (\tilde{p}_{ijk} \times Z_{ijkt}) < \delta, \quad \forall k, t, \quad (13)$$

其中 δ 为充分小的正实数.

由于随机变量总是围绕理论值上下波动, 将工件的实际加工时间 \tilde{p}_{ijk} 用两部分表示, 分别为理论值部分和随机波动部分:

$$\tilde{p}_{ijk} = (1 + \varepsilon \times \theta) \times p_{ijk}, \quad \forall i, j, k, \quad (14)$$

其中, p_{ijk} 表示理论加工时间; 随机变量 θ 表示不确定变量 \tilde{p}_{ijk} 的独立变动程度, 服从参数为 λ 的指数分布; ε ($\varepsilon > 0$) 为波动水平, 用于控制不确定参数的波动幅度.

将式 (14) 代入式 (13) 得

$$(S_{kt} - E_{kt}) + \sum_i \sum_j [(1 + \varepsilon \times \theta) \times p_{ijk} \times Z_{ijkt}] < \delta. \quad (15)$$

整理得

$$\left[S_{kt} - E_{kt} + \sum_{i=1}^n \sum_{j=1}^{n_i} (p_{ijk} \times Z_{ijkt}) \right] + \varepsilon \theta \sum_{i=1}^n \sum_{j=1}^{n_i} (p_{ijk} \times Z_{ijkt}) < \delta. \quad (16)$$

引入参数 κ 来表示约束条件的允许违背程度, 用式 (17) 表示.

$$\Pr \left\{ \left[S_{kt} - E_{kt} + \sum_{i=1}^n \sum_{j=1}^{n_i} (p_{ijk} \times Z_{ijkt}) \right] + \varepsilon \theta \sum_{i=1}^n \sum_{j=1}^{n_i} (p_{ijk} \times Z_{ijkt}) \geq \delta \right\} = \kappa. \quad (17)$$

若令 ϑ 为随机变量 θ 的分位点, 且

$$F_{\theta}(\vartheta) = \Pr(\theta \leq \vartheta) = 1 - \Pr(\theta > \vartheta) = 1 - \kappa. \quad (18)$$

对 $F_\theta(\vartheta)$ 求逆得分位点

$$\vartheta = F_\theta^{-1}(1 - \kappa) = \frac{1}{\lambda} \ln \frac{1}{\kappa}. \quad (19)$$

于是式 (16) 转化为

$$\left[S_{kt} - E_{kt} + \sum_{i=1}^n \sum_{j=1}^{n_i} (p_{ijk} \times Z_{ijkt}) \right] + \varepsilon F_\theta^{-1}(1 - \kappa) \sum_{i=1}^n \sum_{j=1}^{n_i} (p_{ijk} \times Z_{ijkt}) < \delta, \quad (20)$$

其中 $\theta \leq F_\theta^{-1}(1 - \kappa)$ 的概率为 $1 - \kappa$.

用分位点代替随机波动部分, 可将随机变量的 \tilde{p}_{ijk} 转换为确定表达式

$$\tilde{p}_{ijk} = \left(1 + \varepsilon \times \frac{1}{\lambda} \times \ln \frac{1}{\kappa} \right) \times P_{ijk}. \quad (21)$$

综上, 实现了不确定 FJSP 模型到确定型优化调度模型的对等转化. 不失一般性, 不妨假设加工时间扰动部分服从参数为 1 的指数分布, 即 $\lambda = 1$, 并基于此来设计相关的求解算法.

2.4 作业车间调度方案动态调整策略

执行现场工时的不确定性使得调度方案对生产实际现场的指导性能降低, 因此需要对车间调度方案进行快速动态调整, 实现作业计划与执行现场的高度同步. 动态调整主要有移动调整方法和重调度方法. 移动调整体现为作业工序的前后移动调整和插入调整, 重调度方法体现为受影响的可执行工序集的调度排产. 本文将从以下两个方面考虑调度方案的动态调整.

(1) 不确定工时问题的缓解. 在实际生产车间中, 由于各种扰动因素难以确定工时, 本文通过给定波动程度 ε 和约束条件越界限度 κ , 将模型转化为确定性对等模型, 求解该模型可得到一组具有一定程度鲁棒性能的调度方案, 从而提高生产过程的稳定性. 考虑到实时调度的时间复杂性, 本文建立一种缓冲机制, 事先设定工序偏差容忍度, 工序偏差具体表示为式 (22), 这样构建是因为工序加工时间各不一致, 因此相同偏差对各工序影响程度也不一样. 由于调度结构中存在松弛时间, 非关键路径上的扰动可被部分或完全吸收, 而关键路径不存在吸收扰动的松弛时间. 因此本文仅考虑关键路径上的计划开工完工时间偏差, 若关键工序实际开工完工时间小于计划开工完工时间或超过部分的偏差小于 δ (工序调整临界值) 时, 则不改变原始调度方案, 即工序所用机器以及各工序在机器上的加工顺序均不发生改变, 从而在一定程度上减少了作业现场的调整频率. 根据工序实际开工完工时间对工序进行前后移动调整, 修改各工序的计划开始结束加工时间, 如图 2 所示.

$$\delta_{k,t} = \frac{S_{kt} \text{ (or } E_{kt}) - PS_{kt} \text{ (or } PE_{kt})}{\sum_i \sum_j (\tilde{p}_{ijk} \times Z_{ijkt})}, \quad \forall k, t. \quad (22)$$

(2) 不确定工时问题的快速响应. 当关键工序实际开工完工时间超出计划开工完工时间的偏差大于容忍度限度时, 必须快速响应调整避免工时偏差过大对生产车间造成过多经济损失. 车间调度方案具有动态关联性和十分复杂的耦合性, 受到文献 [26] 的启发, 只对部分零件加工工艺路线的片段进行重调度. 采用切分的方式, 首先获取重调度时间点 T_r , T_r 为关键路径上首个执行误差超出容忍度限度的时刻. 对 T_r 时刻所有已经完成和正在加工的工序保持调度方案不变, 将 T_r 时刻后的工序添加至可调度工序集中进行重调度, 根据调度工序和机器约束修改计划加工时间和计划结束时间, 如图 3 所示.

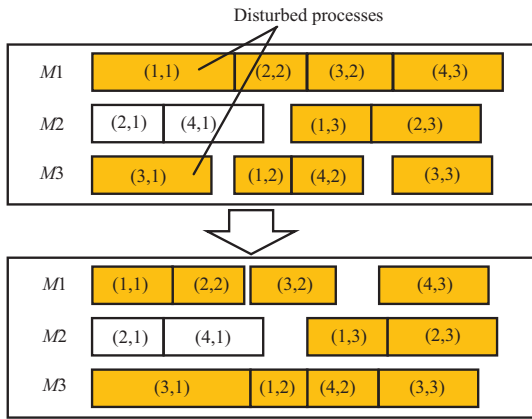


图 2 (网络版彩图) 工序前移、后延调整示意图

Figure 2 (Color online) Process forward and backward adjustments

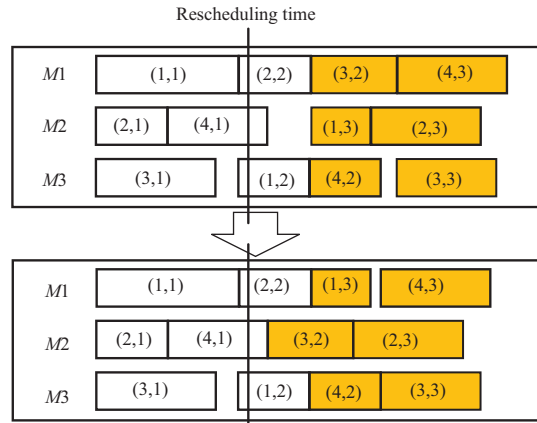


图 3 (网络版彩图) 重调度调整示意图

Figure 3 (Color online) Rescheduling adjustment

3 算法设计

NSGA-II 能够以高效的并行搜索在多维解空间实现全局优化, 操作相比大多数算法更为简便、容易实现, 然而对多目标离散调度问题, (1) NSGA-II 的初始化策略过于随机, 难以保证初始种群的质量; (2) NSGA-II 缺乏有效的局部搜索策略, 难以探索到收敛困难的 Pareto 最优解. 鉴于此, 本文将 VNS 引入 NSGA-II, 提出一种混合多目标遗传算法, 记为 VNS-NSGA-II. 所提算法框架如算法 1 所示. 同 NSGA-II 类似, VNS-NSGA-II 也采用了快速非支配排序、拥挤度、锦标赛选择机制和精英策略, 但 VNS-NSGA-II 会对每一代子种群 Q_t 中的非支配解执行 VNS 策略, 并更新为邻域内较优解, 加速个体收敛至 Pareto 前沿的同时避免算法陷入局部最优.

Algorithm 1 VNS-NSGA-II

Input: G_{\max} (the maximum number of iterations), N (population size).

Output: P (population).

- 1: $P_0 = \text{RHR_Initialization}(N)$; /*Initialize population.*/
 - 2: $P = \text{Non_Domination_Sort_Crowd}(P_0)$; /*Perform fast non-dominated sorting and assign crowding distance on P_0 .*/
 - 3: $G = 1$;
 - 4: **while** $G \leq G_{\max}$ **do**
 - 5: $P' = \text{Tournament_Selection}(P)$;
 - 6: $Q = \text{Variation}(P')$;
 - 7: $U = P \cup Q$;
 - 8: $(F_1, F_2, \dots) = \text{Non_Domination_Sort_Crowd}(U)$;
 - 9: $F'_1 = \text{VNS}(F_1)$; /*Perform variable neighborhood search on non-dominated solutions.*/
 - 10: $U' = (F'_1, F_2, \dots)$; /*Update non-dominated solution.*/
 - 11: $P = \text{Elite_Selection}(U', N)$;
 - 12: $G = G + 1$;
 - 13: **end while**
-

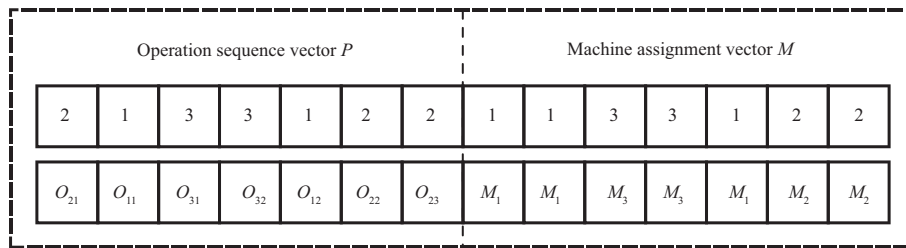


图 4 编码示意图
Figure 4 Encoding diagram

3.1 编码和解码

3.1.1 基于工厂 – 工件号的编码机制

考虑到 FJSP 要解决的两个关键子问题: 工序排序和机器选择, 本文提出基于工件 – 机器号的双链式整数编码机制, 将调度解表示为长度为 $2T_o$ 的工件 – 机器号序列, 它由工件码 P 和机器码 M 组成. P 上的每一个基因都代表一个工件序号, 从左到右读取, 同一工件序号出现 k 次表示该工件包含 k 道工序; M 上的任一基因对应 P 上同一位置的工序所使用的机器号. 如图 4 所示, 第 4 个位置表示的是工件 3 的第 2 道工序, 使用的是第 3 台机器. 这种编码机制两层之间互不影响, 简单灵活, 并且具有很强的并行性.

3.1.2 基于贪婪算法的编码机制

本文采用贪婪插入法^[27], 将染色体解码为主动调度, 具体过程如下.

步骤 1: 从左往右依次读取工序码上的一个基因 O_{ij} , 并且由 M 染色体得到工序 O_{ij} 的加工机器 M_k .

步骤 2: 如果工序 O_{ij} 为机器 M_k 上的第一个操作, 则从该工件上道工序 $O_{i,j-1}$ 的结束时间 $ET_{i,j-1}$ 开始加工; 如果 O_{ij} 同时为该工件第一道工序, 则从 M 的零时刻开始加工; 否则, 记 $ET_{i,j-1}$ 为空闲起点, 在 M 上寻找空闲起点之后的所有空闲时间段, 选择空闲间隔大于等于 p_{ijk} 的时间段 $[TS_h, TE_h], h = 1, 2, \dots$ 根据式 (23) 计算工序 O_{ij} 的最早开始加工时间, 如图 5 所示, 若不存在, 则根据式 (24) 计算工序 O_{ij} 的最早开始加工时间, 如图 6 所示.

$$ST_{ij} = \max\{ET_{i,j-1}, TS_1\}, \tag{23}$$

$$ST_{ij} = \max\{ET_{i,j-1}, LM_k\}, \tag{24}$$

其中 LM_k 为 M_k 上最后一道工序结束时间.

步骤 3: 重复步骤 1 和 2, 直到染色体读取完毕.

3.2 基于复合启发式规则的初始化种群

对于 FJSP, 机器选择比工序排序更为重要, 常见的基于机器选择的初始化方法有: 全局选择 (global selection, GS)、局部选择 (local selection, LS) 和随机选择 (random selection, RS)^[28]. 其中, GS 和 LS 主要考虑机器选择时的负荷问题, 使各台被选择的机器尽量负载均衡, 充分提高机器的利用率; RS 主要考虑尽量使初始种群分散地分布于整个空间, 提高解的多样性. 然而针对 FJSP 问题, 已有初始化策略使得解的质量残次不齐, 较优初始解易导致算法陷入局部最优; 较差初始解导致算法迭代

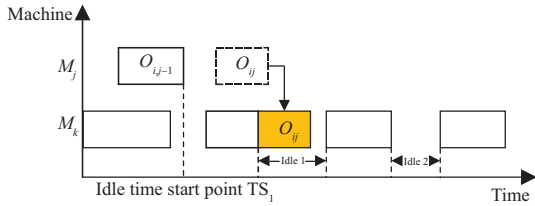


图 5 (网络版彩图) 存在空闲时间段的解码示意图
 Figure 5 (Color online) Decoding with idle time periods

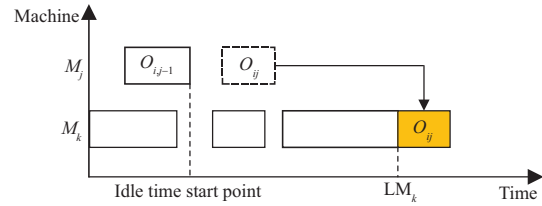


图 6 (网络版彩图) 不存在空闲时间段的解码示意图
 Figure 6 (Color online) Decoding without idle time periods

次数增加. 因此, 本文基于多启发式规则, 提出了复合启发式规则 (recombination heuristic rule, RHR), 该规则既能使加工时间最短的机器优先选用, 又兼顾了机器负载均衡, 在提高初始种群质量的同时也保证了种群的多样性.

RHR 包括: GS, LS, RS, 部分全局选择 (partial global selection, PGS) 和部分局部选择 (partial local selection, PLS). 各部分初始化规则具体如下.

GS: 从未安排机器的工件集中随机选择一个工件, 对被选工件的所有工序按照先后顺序安排加工机器. 每道工序在进行机器选择时, 选择累计负载最小的机器. 重复操作, 直到所有工件被安排完毕.

LS: 依顺序选择待加工工件, 对被选工件的所有工序按照先后顺序安排加工机器. 每道工序选择备选机器集中加工时间最短的机器. 重复操作, 直到所有工件被安排完毕.

RS: 依顺序选择待加工工件, 对被选工件的所有工序按照先后顺序安排加工机器. 对每道工序在进行机器选择时, 随机选择当前工序备选机器的一台. 重复操作, 直到所有工件被安排完毕.

PGS: 从未安排机器的工件集中随机选择一个工件, 对被选工件的所有工序按照先后顺序安排加工机器. 每一道工序在进行机器选择时, 生成一个 0 到 1 之间的随机数, 若小于给定概率, 则选择累计负载最小的机器, 否则随机选择备选机器中的一台. 重复操作, 直到所有工件被安排完毕.

PLS: 依顺序选择待加工工件, 对被选工件的所有工序按照先后顺序安排加工机器. 每一道工序在进行机器选择时, 生成一个 0 到 1 之间的随机数, 若小于给定概率, 则选择备选机器集中加工时间最短的机器, 否则随机选择备选机器中的一台. 重复操作, 直到所有工件被安排完毕.

每个染色体的机器码采用 RHR 规则生成, PGS 和 PLS 策略中随机选择概率均设置为 0.5, 工序码采用完全随机的方式生成. 种群初始化的具体步骤如算法 2 所示, 令 $RHR = [GS, LS, RS, PGS, PLS]$, $RHR(i)$ 表示采用第 i 种策略生成染色体个体.

3.3 多点交叉与变异操作

交叉的目的是组合父代个体信息产生新的个体, 在尽量不破坏已有模式的基础上对解空间进行有效搜索. NSGA-II 的性能很大程度上依赖于所使用的交叉操作, 它决定了 NSGA-II 的全局搜索能力. 为了保证交叉操作后产生的是可行解, 工序码采用 IPOX 交叉^[29], 机器码采用多点交叉^[30].

变异操作通过对染色体进行较小扰动来生成新的个体, 增加种群多样性, 在一定程度上影响着 NSGA-II 算法的局部搜索能力. 利用邻域结构扩大搜索空间, 搜索得到邻域内的较优解, 优化局部最优解. 针对工序码采用基于邻域搜索的变异方式^[11], 定义邻域为工序码中代表不同工件的 τ 个不同位置的基因通过位置交换所构成的所有可能组合. 定义评价函数为

$$f = f_1 + f_2 + f_3. \tag{25}$$

Algorithm 2 RHR_Initialization**Input:** N (population size), p_i (the ratio of the i -th strategy), $i = 1, 2, \dots$, and $\sum_{i=1}^5 p_i = 1$.**Output:** initialpop (initial population).

```

1:  $i = 1$ ; initialpop =  $\emptyset$ ;
2: while  $i \leq 5$  do
3:    $j = 0$ ;
4:   subpopulation =  $\emptyset$ ;
5:   while  $j \leq \text{pop} * p_i$  do
6:     subpopulation = subpopulation  $\cup$  {RHR( $i$ )}; /*Add the chromosome generated by the  $i$ -th strategy to
       subpopulation.*/
7:      $j = j + 1$ ;
8:   end while
9:   initialpop = initialpop  $\cup$  {subpopulation};
10:   $i = i + 1$ ;
11: end while

```

在工序码上随机选择 τ 个位置, 通过评价函数找出当前邻域内的最优解, 并更新当前解. 改变 τ 值, 改变邻域结构, τ 取值范围为 2~4. 针对机器码的变异为: 随机选取两道用于机器变异的工序, 然后在这两道工序余下可选机器集中任意选取一台, 并将机器码上对应位置的基因更新为所选机器号.

3.4 变邻域搜索策略

基于析取图理论产生的关键路径邻域搜索是一种局部高效的求解方法, 其核心思想在于最大化利用机器空闲时间. 典型的已有邻域结构有 Nowicki 等^[31]、Vazacopoulos 等^[32]、Zhang 等^[33] 分别提出的 N5, N6, N7 结构. 在邻域结构的理论研究方面, 具有代表性的是 Nowicki 等^[31]、van Laarhoven 等^[34]、Matsuo 等^[35] 提出的相关工序移动交换性质, 有效指导了邻域结构的设计. Liu 等^[36] 在参与移动工序数量与移动方式上对邻域结构部分性质进一步拓展, 具体如下.

推论 1: 非关键工序内部的任意移动不能减小最大完工时间.

推论 2: 关键路径首块中块尾工序的变化可能减小最大完工时间, 非块尾工序的任意移动不能减小最大完工时间.

推论 3: 关键路径尾块中块首工序的变化可能减小最大完工时间, 非块首工序的任意移动不能减小最大完工时间.

推论 4: 关键路径中间块中块首、块尾工序的变化可能减小最大完工时间, 非块首、块尾工序的任意移动不能减小最大完工时间.

邻域结构设计是变邻域搜索算法的关键部分, 长期试验发现, 有效同机器关键工序移动往往存在于关键工序前移中^[37], 为此, 本文针对关键工序前移设计邻域结构如图 7 所示, 固定各关键块块首, 首块中块尾工序移至块首/块内工序之前, 尾块中块内/块尾工序移至块首工序之前, 中间块中块内工序移至块首之前、块尾工序移至块首/块内工序之前. 此外基于同机器关键工序交换设计邻域结构如图 8 所示, 首块中交换块尾工序与块首/块内工序, 尾块中交换块首工序与快内/块尾工序, 中间块交换块首工序与块内工序或交换块首/块内工序与块尾工序.

由于同机器移动或交换关键工序并不改变工序在机器上分配的结果, 因此同机器移动或交换工序可有效降低目标值 f_1 , 但对目标值 f_2, f_3 并无影响. 为了在进化过程中能同时优化目标值 f_2 与 f_3 , 针对机器选择链进行操作, 每道工序以 50% 的概率按照机器负载最小或加工时间最小原则从备选加工机器集中选择加工机器.

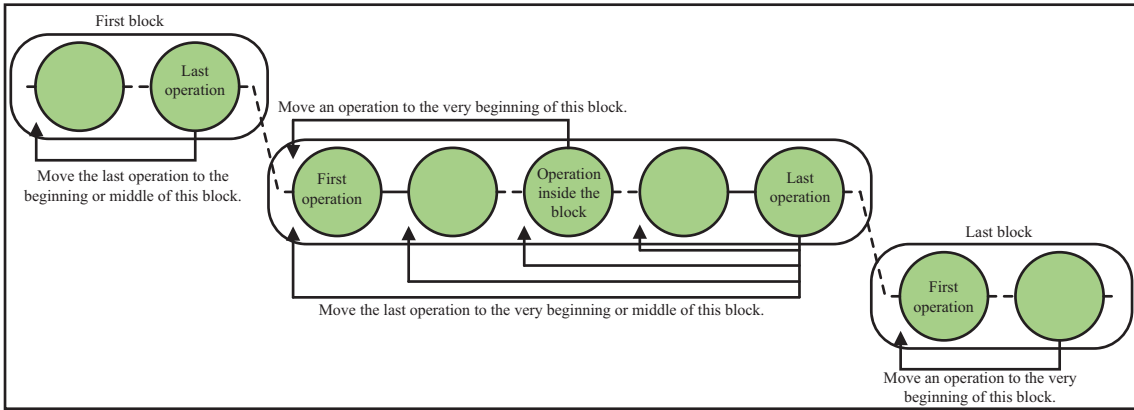


图 7 (网络版彩图) 关键工序前移邻域结构

Figure 7 (Color online) Neighborhood structure of key-process forward-moving

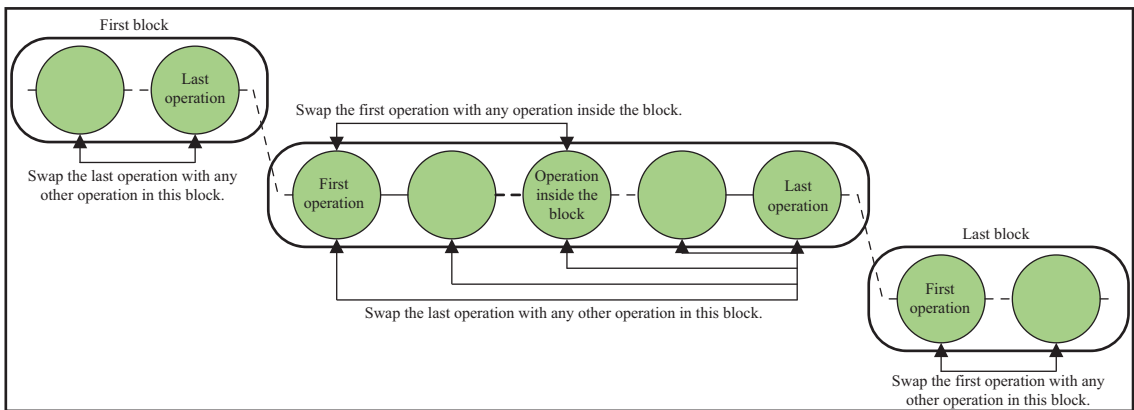


图 8 (网络版彩图) 关键工序交换邻域结构

Figure 8 (Color online) Neighborhood structure of key-process exchanging

为加快算法的收敛速度同时避免种群陷入局部最优状态, 本文对每一次迭代生成的子种群 Q (算法 1 第 8 行) 进行快速非支配排序, 将得到的非支配解集 F_1 执行 VNS 操作. VNS 算法的具体操作过程如算法 3 所示. $neighbor_variation(x)$ 表示生成 x 的一个邻域解, 具体过程如算法 4 所示. 对 F_1 中的每一个个体 x_j ($j = 1, 2, \dots, |F_1|$), 使用函数 $neighbor_variation(x_j)$ 生成 x_j 的 $neighbor_num$ 个邻域解, 并将所有邻域解存放至临时解集 S 中. 最后对 S 进行快速非支配排序和拥挤度计算, 并基于精英策略将 F_1 更新为邻域中的最优解集.

3.5 算法时间复杂度分析

设 T 为目标个数, N 为种群规模, $neighbor_num$ 为个体产生邻域解个数. 对种群进行 Pareto 非支配排序需要进行 $O(TN^2)$ 次比较 (算法 1 第 8 行), 计算种群中个体的拥挤度距离的时间复杂度为 $O(TN \log N)$ (算法 1 第 8 行). 使用 VNS 策略更新非支配解集 F_1 包括 3 个步骤 (算法 1 第 9 行): 首先生成 F_1 的邻域 (算法 3 第 3~8 行), 时间复杂度为 $O(N \times neighbor_num)$; 其次进行非支配排序和拥挤度计算 (算法 3 第 9 行), 由于最坏条件下, 所有个体均为非支配解, 此时所需要的时间复杂度为 $O(TN^2)$; 最后采用精英策略选择 $|F_1|$ 个解 (算法 3 第 10 行), 时间复杂度为 $O(N \log N)$, 故执行一次

Algorithm 3 VNS**Input:** F_1 (non-dominated solution set), neighbor_num (the number of neighborhood solutions generated by one solution).**Output:** F'_1 (the updated non-dominated solution set).

```

1:  $S = \emptyset$ ;
2:  $n = 1$ ;
3: for each  $x_j \in F_1$  do
4:   while  $n \leq \text{neighbor\_num}$  do
5:      $S \cup \text{neighbor\_variation}(x_j)$ ;
6:      $n = n + 1$ ;
7:   end while
8: end for
9:  $S' = \text{Non\_Domination\_Sort\_Crowd}(S)$ ;
10:  $F'_1 = \text{Elite\_Selection}(S', |F_1|)$ ;

```

Algorithm 4 neighbor_variation**Input:** x .**Output:** better_x (a better neighborhood solution of x).

```

1: better_x = x;
2: Find key processes and determine process blocks; each process block contains at least 2 key processes;
3:  $x' = \text{Neighbor\_Search}(x)$ ; /*Randomly choose a neighborhood structure to generate neighborhood solution.*/
4: if  $x' \preceq \text{better\_x}$  then
5:   better_x =  $x'$ ; /*If  $x'$  dominates better_x or  $x'$  and better_x are equivalent, then update better_x.*/
6: end if

```

VNS 所需要的时间复杂度为 $O(TN^2)$. 因此最坏情况下的 VNS-NSGA-II 迭代一次所需要的整体时间复杂度为 $O(TN^2)$.

4 仿真实验及结果分析

本文所提算法基于 MATLAB2021 编程实现, 在 Windows10.0 系统, 处理器 Intel Pentium CPU 4415U, 2.30 GHz, 4 GB RAM 的 PC 机上运行. 为适应不同算例规模, 将种群规模设为 $N = 2 \times m \times n$, 最大迭代次数 $G_{\max} = m \times n$. 为均衡使用各种初始化策略, RHR 的 5 种启发式规则比例为 0.2:0.2:0.2:0.2:0.2. 交叉和变异概率分别设置为 $p_c = 0.8$ 和 $p_m = 0.1$ [38]. 为平衡算法效率与邻域搜索, 非支配解生成邻域个数 neighbor_num = 2.

4.1 算法性能测试

为验证算法的性能, 本文使用两个国际通用的标准案例集进行测试并与其他成熟算法进行对比. 第一个案例集为 Kacem [39], 包含 4 个问题实例, 分别为 $4 \times 5, 8 \times 8, 10 \times 10, 10 \times 15$. 第二个案例集为 Brandimarte [40] 的 10 个问题实例: MK01-MK10.

首先测试 Kacem 案例集, 并将测试结果与 4 个成熟算法进行对比: MOGA [38], TLHGAPSO [41], SM [42] 以及 NSGA-II, 各算法求解结果如表 1 所示, 黑体加粗表示该解被对比算法中的相应解支配. 为更直观描述本文算法优越性, 对表 1 结果进行综合统计分析, 统计结果如表 2 所示, 其中 s 表示被对比算法支配的解的数量, S 表示生成解的数量.

分析表 2 可知, 在 Kacem 的 4 个问题实例上, TLHGAPSO, SM 算法生成最优解的数量均少

表 1 Kacem 案例各算法结果对比

Table 1 Comparison of the results of each algorithm in the Kacem case

Instance	Objective	MOGA [38]				TL_HGAPSO [41]			SM [42]			NSGA-II [18]				VNS-NSGA-II			
4 × 5	f_1	11	11	12	-	-	-	-	11	12	-	13	11	12	-	11	12	13	11
	f_2	10	9	8	-	-	-	-	9	8	-	7	11	8	-	10	8	7	9
	f_3	32	34	32	-	-	-	-	34	32	-	33	33	32	-	32	32	33	34
8 × 8	f_1	15	15	16	-	14	15	16	14	16	16	15	16	-	-	16	15	16	-
	f_2	11	12	13	-	12	12	13	12	13	11	12	13	-	-	13	12	11	-
	f_3	81	75	73	-	77	75	73	77	73	77	75	73	-	-	73	75	77	-
10 × 10	f_1	8	7	8	7	7	8	-	8	7	8	7	8	8	-	7	8	-	-
	f_2	5	6	7	5	6	7	-	5	6	7	6	7	5	-	6	7	-	-
	f_3	42	42	41	45	42	41	-	42	42	41	42	41	45	-	42	41	-	-
10 × 15	f_1	11	12	11	-	12	-	-	11	-	-	13	13	13	14	11	12	-	-
	f_2	11	10	10	-	11	-	-	11	-	-	11	12	13	12	11	10	-	-
	f_3	91	95	98	-	91	-	-	91	-	-	96	95	94	93	91	93	-	-

表 2 Kacem 案例各算法结果综合分析对比

Table 2 Comprehensive analysis and comparison of the results of various algorithms in the case of Kacem

Instance	MOGA	TL_HGAPSO	SM	NSGA-II	VNS-NSGA-II
	s, S	s, S	s, S	s, S	s, S
4 × 5	0, 3	-	0, 2	1, 3	0, 4
8 × 8	0, 3	0, 3	0, 3	0, 2	0, 3
10 × 10	0, 4	0, 2	0, 3	1, 3	0, 2
10 × 15	1, 3	1, 1	0, 1	4, 4	0, 2
Tatal	1, 13	1, 6	0, 9	6, 12	0, 11

于 VNS-NSGA-II, 且 TL_HGAPSO 中存在一个解被支配. MOGA, NSGA-II 算法生成最优解的数量多于 VNS-NSGA-II 所得最优解的数量, 但 MOGA 存在一个解被支配, NSGA-II 存在 6 个解被支配, VNS-NSGA-II 不存在被支配的解. 综上, VNS-NSGA-II 算法较对比算法性能更优.

为进一步说明所提算法的优越性, 采用综合评价指标反向世代距离 IGD (inverted generational distance) 进行分析, IGD 计算公式如式 (26) 所示:

$$IGD(P, P^*) = \frac{1}{|P^*|} \sum_{x \in P^*} \min_{y \in P} d(x, y), \quad (26)$$

其中 P^* 表示由最优解构成的 Pareto 前沿, $|P^*|$ 表示 Pareto 前沿最优解的个数, P 表示算法实际所求最优解, $d(x, y)$ 表示 x 与 y 之间的欧氏距离. 由于车间调度测试集真实 Pareto 前沿未知, 所以需要使用某些求解到的解代替计算 IGD 值所需要的 P^* . 本文将所有对比算法得到的非支配解集进行混合得到混合解集, 选取混合解集中的非支配解作为 P^* . 计算 IGD 值时, 将混合解集中解的目标值做归一化处理, 处理方法如式 (27) 所示:

$$\bar{f}(x) = \frac{f_i(x) - f_i^{\min}}{f_i^{\max} - f_i^{\min}}. \quad (27)$$

IGD 值越小, 说明解越靠近真实 Pareto 前沿且分布越均匀.

表 3 Brandimarte 案例各算法 IGD 值对比

Table 3 Comparison of IGD values of various algorithms in the case of Brandimarte

Instance	MOGA	TL_HGAPSO	SM	NSGA-II	VNS-NSGA-II
MK01 (10 × 6)	0.202369	0.413167	0.768828	0.033192	0.000000
MK02 (10 × 6)	0.031906	0.199457	0.909655	0.080064	0.077162
MK03 (15 × 8)	0.301833	0.362636	0.481058	0.206467	0.086097
MK04 (15 × 8)	0.261854	0.432673	0.487852	0.082940	0.045031
MK05 (15 × 4)	0.231697	0.317021	0.982452	0.093585	0.009850
MK06 (10 × 15)	0.199224	0.228099	0.664739	0.054457	0.029623
MK07 (20 × 5)	0.118652	0.206680	0.864035	0.124210	0.028251
MK08 (20 × 10)	0.139014	0.598889	0.729840	0.033116	0.033116
MK09 (20 × 10)	0.274646	0.459795	0.504908	0.054053	0.014507
MK10 (20 × 15)	0.108898	0.157308	0.457295	0.197294	0.068434

表 3 为 MOGA, TL_HGAPSO, SM, NSGA-II 以及 VNS-NSGA-II 算法的 IGD 值, 其中从表中可以看出, VNS-NSGA-II 在 9 个算例上获得最小 IGD 值, 且在 MK01 算例上完全覆盖 P^* , MOGA 在 MK02 算例上获得最小 IGD 值, NSGA-II 与 VNS-NSGA-II 在 MK08 算例上同时取得最小 IGD 值, TL_HGAPSO 与 SM 不曾获得最小 IGD 值. 因此可得出结论, VNS-NSGA-II 较对比算法更优越.

以 MK01 与 MK05 为例, MOGA, TL_HGAPSO, NSGA-II 和 VNS-NSGA-II 产生最佳 IGD 值的非支配解分布如图 9 和 10 所示, 其中红色“+”表示近似 P^* , 绿色实心圆表示算法所求非支配解. 从图中可以看出, VNS-NSGA-II 解的分布性最好, 且生成非支配解的数量最多; NSGA-II 所得解分布性良好但收敛性较差; MOGA 与 TL_HGAPSO 分布性较差, 且仅能求得 P^* 上局部解.

4.2 参数敏感性与动态调度分析

当考虑车间系统中不确定因素对工序加工时间的影响时, 定义最优适应度 fit_{best} 为

$$\text{fit}_{\text{best}} = \min \text{fit}_i, \quad i = 1, \dots, N, \quad (28)$$

其中 fit_i 为算法所得第 i 个非支配解的适应度, 由式 (25) 定义, N 为非支配解个数. 提出随机工时条件下 $\tilde{\text{fit}}_{\text{best}}$ 与理想 fit_{best} 的偏差值为系统风险评价指标 (risk measure, RM), 表示为

$$\text{RM} = \tilde{\text{fit}}_{\text{best}} - \text{fit}_{\text{best}}. \quad (29)$$

以最小化 RM 指标为目标函数设计实验, 进行参数敏感性分析. 本文建立的确定性对等模型中的重要控制参数包括随机变量独立变动程度 ε 和约束条件允许违背程度 κ . 不同的控制参数对应不同的对等确定性模型, 为研究不同控制参数对确定性模型所得最优调度方案性能的影响, 实验采用 4×5 的 Kacem 算例. 待测试问题的随机变量波动程度的取值 $E = \{0.005, 0.010, 0.015, 0.020, 0.025\}$, 约束条件允许违背程度的取值 $K = \{5\%, 10\%, 15\%, 20\%, 25\%, 30\%\}$, 所以, 共有 30 组不同的组合实验 ($|E| \times |K|$). 每组实验使用 VNS-NSGA-II 算法连续运行 10 次求解对等模型得到一组平均 RM 值. 实验结果如图 11 所示, 分析图 11 可知, (1) 对等问题的最优解总是大于确定条件下的最优解; (2) 当 κ 值确定时, RM 均值随着 ε 的增大而逐渐增大, 即工时扰动程度 ε 越大, 对解的影响越大, 所求目标函数值越偏离理论最优解; (3) 当 ε 值确定时, RM 均值随着 κ 的增大而逐渐减小, 即工时扰动对目标函

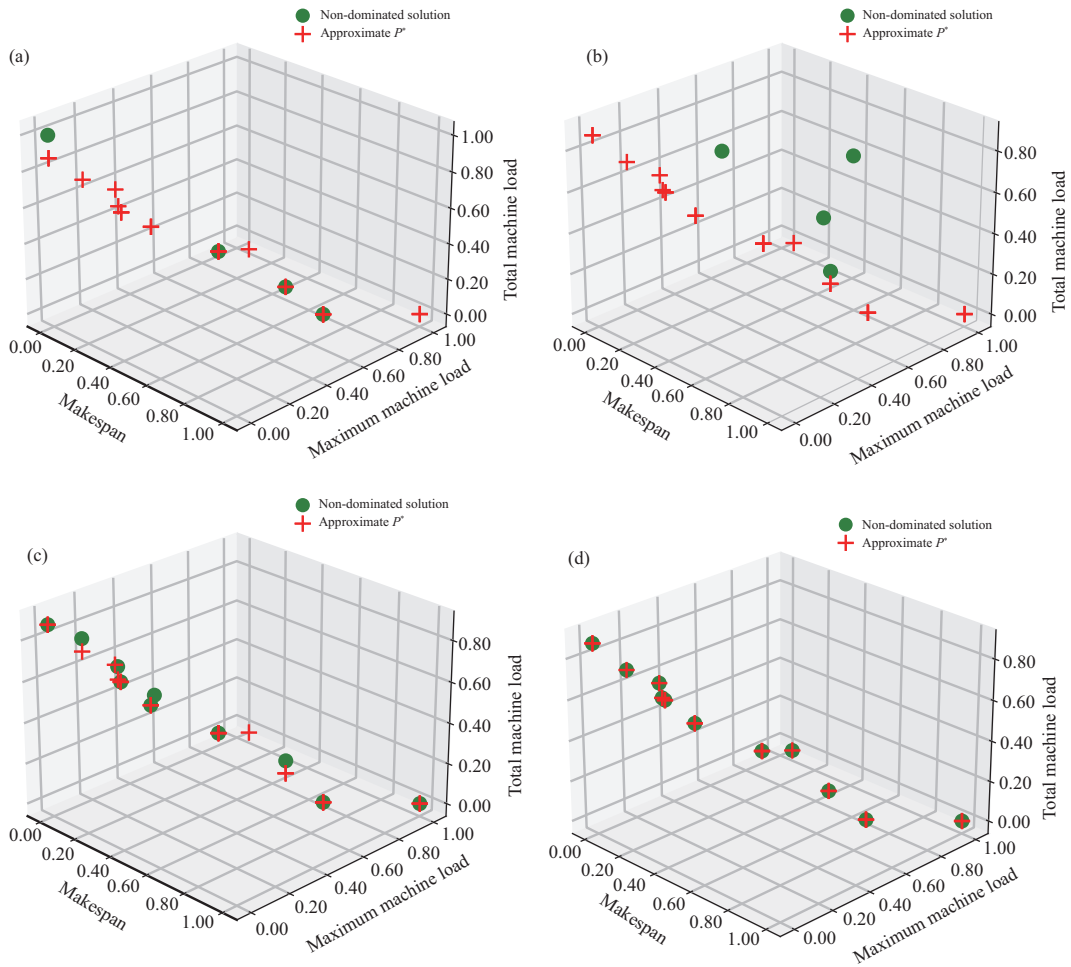


图 9 (网络版彩图) MK01 各算法解空间分布对比

Figure 9 (Color online) Comparison of the solution space distribution of various algorithms on MK01. (a) MOGA; (b) TL-HGAPSO; (c) NSGA-II; (d) VNS-NSGA-II

数值的影响随着约束条件不可行限度的增加而减小, 说明约束条件越界限度与性能损失之间存在一定的折中; (4) 当 ε 越小, κ 越大时, RM 均值趋于 0, 此时吸收不确定扰动所需的冗余较小, 调度方案性能损失也较小.

下面对 4×5 的 Kacem 算例进行仿真实验描述动态调整过程. 设定偏差容忍度 $\delta = 15\%$, 波动水平 $\varepsilon = 0.01$, 约束条件越界限度 $\kappa = 0.05$. 根据转换后的对等模型确定原始调度方案如图 12 所示, 其中 (i, j) 表示工序 O_{ij} . 关键工序包括 $\{O_{11}, O_{12}, O_{13}, O_{33}, O_{34}\}$. 使用 Monte Carlo 方法根据工序加工时间概率分布进行随机采样, 生成一组随机场景进行仿真实验. 对整个生产过程进行动态调度调整, 最终调整结果如图 13 所示, 工序 O_{12} 的实际完工时间为 7.05, 晚于计划完工时间 5.15, 该道工序的总加工时间为 6.05, 实际执行偏差为 31.4% 大于容忍度 15%. 因此设定重调度时间点 T_r 为 7.05, 对该时间点之后的所有工序块使用 VNS-NSGA-II 算法进行重新作业排产安排, 后续生产过程将按照调整后的方案继续进行. 整个生产过程中发生过一次重调度, 其余工序均为保持调度顺序不变的前移、后延调度调整.

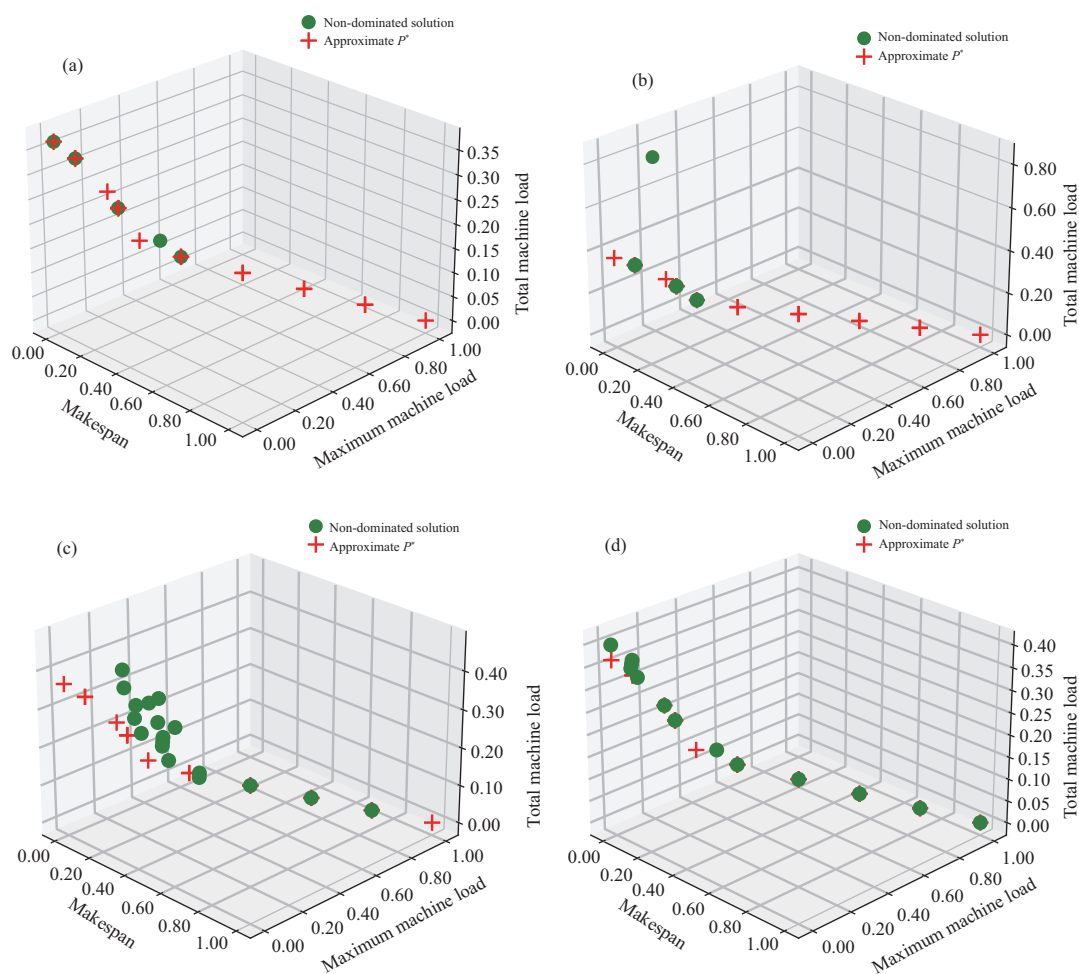


图 10 (网络版彩图) MK05 各算法解空间分布对比

Figure 10 (Color online) Comparison of the solution space distribution of various algorithms on MK05. (a) MOGA; (b) TL-HGAPSO; (c) NSGA-II; (d) VNS-NSGA-II

5 结论

本文针对随机工时下的多目标 FJSP 问题, 建立工时波动服从指数分布的不确定车间调度模型, 利用机会约束规划理论处理工时不确定性, 并转换为确定性对等调度模型. 提出混合遗传算法 VNS-NSGA-II, 算法采用双链式编码以及贪婪插入法解码规则, 设计 RHR 种群初始化方法产生较优初始种群, 加快算法收敛速度的同时保证了了解的多样性. 在该算法中, 以 NSGA-II 为算法主体, 负责解空间的全局搜索, 以 VNS 算法为辅, 负责解的局部搜索, 从而平衡算法的广度搜索能力与深度搜索能力. 最后通过 Kacem 和 Brandimarte 标准算例的数值仿真以及与多种代表算法进行统计比较. 仿真结果表明, VNS-NSGA-II 算法的性能相比传统 NSGA-II 取得显著提升, 能得到各类基准测试算例质量较好且数量较多的非支配解. 在加工时间存在不确定扰动情况下, VNS-NSGA-II 首先求得对等模型的最优解作为预调度方案, 并在实际生产中采用工序移动调整和重调度方法对作业排产方案进行动态调整. 需要注意的是, ε 越大, κ 越小, 对等模型对应的调度方案鲁棒性能越强, 但可能会造成设备利用率低, 经济性变差, 甚至可能无法按时交付订单. 因此在实际应用时, 决策者应设置合理的波动幅度和约

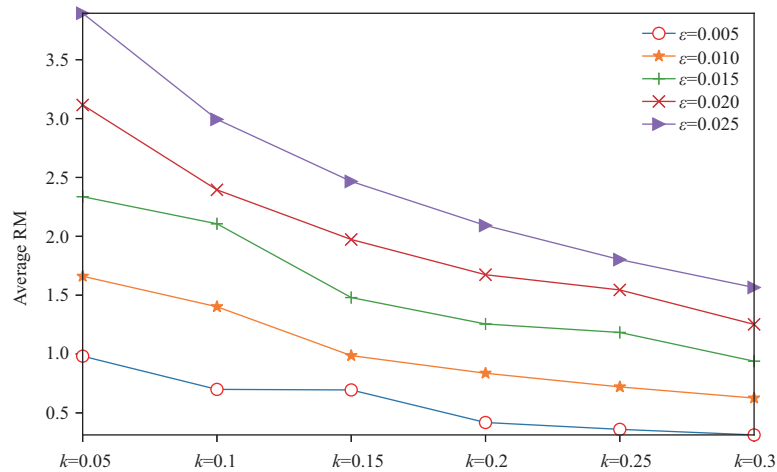


图 11 (网络版彩图) 不同控制参数对 RM 的影响

Figure 11 (Color online) The influence of different control parameters on RM

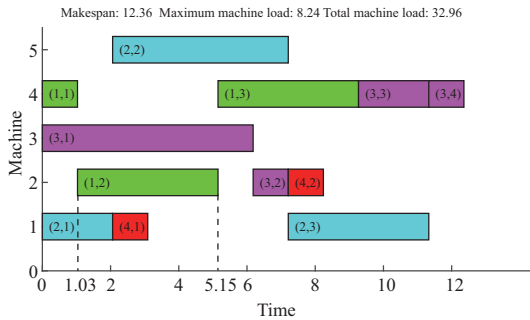


图 12 (网络版彩图) 原始调度方案

Figure 12 (Color online) Original schedule

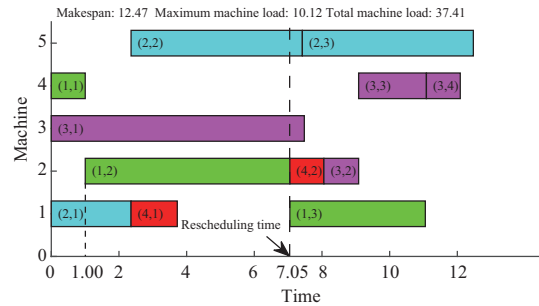


图 13 (网络版彩图) 动态调整后的调度方案

Figure 13 (Color online) Dynamically adjusted schedule

束条件越界限度以平衡系统运行经济性、可靠性之间的关系. 本文提出的 VNS-NSGA-II 算法仅考虑 3 个目标以及较小规模的调度问题, 故后续的主要工作为研究大规模以及超多目标的随机调度问题.

参考文献

- 1 Rakovitis N, Li D, Zhang N, et al. Novel approach to energy-efficient flexible job-shop scheduling problems. *Energy*, 2021, 238: 121773
- 2 Zhang H L, Xu G J, Pan R L, et al. A novel heuristic method for the energy-efficient flexible job-shop scheduling problem with sequence-dependent set-up and transportation time. *Eng Optimization*, 2022, 54: 1646-1667
- 3 Li J Q, Du Y, Tian J, et al. An artificial bee colony algorithm for flexible job shop scheduling with transportation resource constraints. *Acta Electron Sin*, 2021, 49: 324-330 [李俊青, 杜宇, 田杰, 等. 带运输资源约束柔性作业车间调度问题的人工蜂群算法. *电子学报*, 2021, 49: 324-330]
- 4 Zheng Y L, Lei D M, Zheng Q X. Novel artificial bee colony algorithm for solving many-objective scheduling. *Comput Sci*, 2020, 47: 186-191 [郑友莲, 雷德明, 郑巧仙. 求解高维多目标调度的新型人工蜂群算法. *计算机科学*, 2020, 47: 186-191]
- 5 Lei D M, Zheng Y L, Guo X P. A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption. *Int J Production Res*, 2017, 55: 3126-3140

- 6 Gao K, Yang F, Zhou M C, et al. Flexible job-shop rescheduling for new job insertion by using discrete Jaya algorithm. *IEEE Trans Cybern*, 2018, 49: 1944–1955
- 7 Goren S, Sabuncuoglu I. Optimization of schedule robustness and stability under random machine breakdowns and processing time variability. *IIE Trans*, 2009, 42: 203–220
- 8 Yang H A, Wang Z F, Lv Y Y, et al. Interval number solving method for job-shop scheduling problem with processing time variability. *Comput Integr Manuf Sys*, 2014, 20: 2231–2240 [杨宏安, 王周锋, 吕阳阳, 等. 工序加工时间不确定条件下作业车间调度问题的区间数求解方法. *计算机集成制造系统*, 2014, 20: 2231–2240]
- 9 Sun L, Lin L, Li H J, et al. Cooperative co-evolution algorithm with an MRF-based decomposition strategy for stochastic flexible job shop scheduling. *Mathematics*, 2019, 7: 318
- 10 Horng S C, Lin S S. Two-stage bio-inspired optimization algorithm for stochastic job shop scheduling problem. *Int J Simul Syst Sci Tech*, 2015, 16: 1–8
- 11 Tang Q H, He M, He X X, et al. Robust optimization scheduling of flexible job shops under stochastic processing times. *Comput Integr Manuf Sys*, 2015, 21: 1002–1012 [唐秋华, 何明, 何晓霞, 等. 随机工时下柔性加工车间的鲁棒优化调度方法. *计算机集成制造系统*, 2015, 21: 1002–1012]
- 12 Graves S C, Kan A H G R, Zipkin P H. *Logistics of Production and Inventory*. Amsterdam: North Holland, 1993. 4: 3–760
- 13 Law A M, Kelton W D. *Simulation modeling and analysis*. *J Am Stat Assoc*, 1991. doi: 10.2307/2288169
- 14 Chen L, Yan C P, Chen J L, et al. Dynamic scheduling of flexible job shop based on deep Q-learning neural network and quantum genetic algorithm. *J Chongqing Univ*, 2022, 45: 40–54 [陈亮, 阎春平, 陈建霖, 等. 基于深度学习神经网络和量子遗传算法的柔性作业车间动态调度. *重庆大学学报*, 2022, 45: 40–54]
- 15 Ren J F, Ye C M. Method to solve Job-Shop scheduling problem using deep recurrent neural network model with embedded pointer network. *Appl Res Comput*, 2021, 38: 120–124, 128 [任剑锋, 叶春明. 嵌入指针网络的深度循环神经网络模型求解作业车间调度问题. *计算机应用研究*, 2021, 38: 120–124, 128]
- 16 Li B S, Ye C M. Job shop scheduling problem based on deep reinforcement learning. *Comput Eng Appl*, 2021, 57: 248–254 [李宝帅, 叶春明. 深度强化学习算法求解作业车间调度问题. *计算机工程与应用*, 2021, 57: 248–254]
- 17 Chaudhry I A, Khan A A. A research survey: review of flexible job shop scheduling techniques. *Intl Trans Op Res*, 2016, 23: 551–591
- 18 Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Computat*, 2002, 6: 182–197
- 19 Wang Y K, Liu Y B, Wu Y M, et al. Improved NSGA-II algorithm to solve energy-saving scheduling problem of flexible job shop considering transportation constraints. *Comput Integr Manuf Sys*, 2021 [王亚昆, 刘应波, 吴永明, 等. 改进 NSGA-II 算法求解考虑运输约束的柔性作业车间节能调度问题. *计算机集成制造系统*, 2021]. <http://kns.cnki.net/kcms/detail/11.5946.tp.20210912.2002.004.html>
- 20 Seng D W, Li J W, Fang X J, et al. Low-carbon flexible job-shop scheduling based on improved nondominated sorting genetic algorithm-II. *Int J Simul Model*, 2018, 17: 712–723
- 21 Tan W, Yuan X, Wang J, et al. A fatigue-conscious dual resource constrained flexible job shop scheduling problem by enhanced NSGA-II: an application from casting workshop. *Comput Industrial Eng*, 2021, 160: 107557
- 22 Gao K Z, Suganthan P N, Pan Q K, et al. Pareto-based grouping discrete harmony search algorithm for multi-objective flexible job shop scheduling. *Inf Sci*, 2014, 289: 76–90
- 23 Gong G, Deng Q, Chiong R, et al. An effective memetic algorithm for multi-objective job-shop scheduling. *Knowledge-Based Syst*, 2019, 182: 104840
- 24 Zhao S K. Hybrid algorithm based on improved neighborhood structure for flexible job shop scheduling. *Comput Integr Manuf Sys*, 2018, 24: 3060–3072 [赵诗奎. 柔性作业车间调度的改进邻域结构混合算法. *计算机集成制造系统*, 2018, 24: 3060–3072]
- 25 Charnes A, Cooper W W. Chance-constrained programming. *Manage Sci*, 1959, 6: 73–79
- 26 Ding L, Wang A M, Ning R X. Job-Shop scheduling technology with uncertain processing time. *Comput Integr Manuf Sys*, 2010, 16: 11 [丁雷, 王爱民, 宁汝新. 工时不确定条件下的车间作业调度技术. *计算机集成制造系统*, 2010, 16: 11]
- 27 Cheng X, Kong Q, Wu Q. Hybrid algorithm for job-shop scheduling problem. In: *Proceedings of the 4th World Congress on Intelligent Control and Automation*, 2002. 3: 1739–1743

- 28 Zhang C Y, Rao Y Q, Li P G, et al. An improved genetic algorithm for solving job shop scheduling problems. *Comput Integr Manuf Sys*, 2004, 10: 966–970 [张超勇, 饶运清, 李培根, 等. 求解作业车间调度问题的一种改进遗传算法. *计算机集成制造系统*, 2004, 10: 966–970]
- 29 Wang X J. Research on methods for multi-objective flexible job-shop scheduling problem. Dissertation for Ph.D. Degree. Wuhan: Huazhong University of Science and Technology, 2011 [王晓娟. 多目标柔性作业车间调度方法研究. 博士学位论文. 武汉: 华中科技大学, 2011]
- 30 Zhang C Y, Rao Y Q, Li P G, et al. Bilevel genetic algorithm for the flexible job-shop scheduling problem. *Chin J Mech Eng-En*, 2007: 119–124 [张超勇, 饶运清, 李培根, 等. 柔性作业车间调度问题的两级遗传算法. *机械工程学报*, 2007, 43: 119–124]
- 31 Nowicki E, Smutnicki C. A fast taboo search algorithm for the job shop problem. *Manage Sci*, 1996, 42: 797–813
- 32 Vazacopoulos B A. Guided local search with shifting bottleneck for job shop scheduling. *Manage Sci*, 1998, 44: 262–275
- 33 Zhang C Y, Li P G, Guan Z L, et al. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. *Comput Operations Res*, 2007, 34: 3229–3242
- 34 van Laarhoven P J M, Aarts E H L, Lenstra J K. Job shop scheduling by simulated annealing. *Operations Res*, 1992, 40: 113–125
- 35 Matsuo H, Suh C J, Sullivan R S. A controlled search simulated annealing method for the single machine weighted tardiness problem. *Ann Oper Res*, 1989, 21: 85–108
- 36 Liu G, Wang Y, Zhang F. Convergence analysis for solving job shop scheduling problem based on critical path. *Comput Integr Manuf Sys*, 2014, 20: 10 [刘刚, 王瑛, 张发. 基于关键路径求解作业车间调度问题的收敛性分析. *计算机集成制造系统*, 2014, 20: 10]
- 37 Wang J H, Li Y L, Liu Z W, et al. Hybrid evolutionary algorithm of precise neighborhood structure for flexible job shop scheduling. *J Tongji Univ Nat Sci*, 2021, 49: 440–448 [王家海, 李莹力, 刘铮玮, 等. 柔性作业车间调度的精确邻域结构混合进化算法. *同济大学学报 (自然科学版)*, 2021, 49: 440–448]
- 38 Wang X, Gao L, Zhang C, et al. A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem. *Int J Adv Manuf Technol*, 2010, 51: 757–767
- 39 Kacem I, Hammadi S, Borne P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Trans Syst Man Cybern C*, 2002, 32: 1–13
- 40 Brandimarte P. Routing and scheduling in a flexible job shop by tabu search. *Ann Oper Res*, 1993, 41: 157–183
- 41 Huang X, Guan Z, Yang L. An effective hybrid algorithm for multi-objective flexible job-shop scheduling problem. *Adv Mech Eng*, 2018, 10: 168781401880144
- 42 Xing L N, Chen Y W, Yang K W. An efficient search method for multi-objective flexible job shop scheduling problems. *J Intell Manuf*, 2009, 20: 283–293

An evolutionary algorithm for the multi-objective flexible job shop scheduling problem with uncertain processing time

Xiaoyu ZHONG¹, Yuyan HAN², Xiangjuan YAO^{1,4*}, Dunwei GONG^{3*} & Yongzheng SUN¹

1. School of Mathematics, China University of Mining and Technology, Xuzhou 221116, China;

2. School of Computer Science, Liaocheng University, Liaocheng 252000, China;

3. School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China;

4. State Key Laboratory of New Computer Software Technology, Nanjing University, Nanjing 210023, China

* Corresponding author. E-mail: yaoxj@cumt.edu.cn, dwgong@vip.163.com

Abstract In practical industrial production, the complexity and uncertainty of the scheduling environment greatly increase the difficulty in solving the scheduling problem. To resolve the flexible job shop scheduling problem with uncertain processing time, two uncertain parameters are used to describe the fluctuation degree of random processing time and the allowable degree of constraint violation. In addition, a multi-objective flexible job shop scheduling mathematical model is constructed, the processing time fluctuation of which obeys exponential distribution. Based on the chance-constrained programming theory, the uncertain scheduling problem is transformed into a flexible job shop scheduling problem with a determined processing time, and a scheduling scheme with robust performance (to a certain extent) is obtained. During its execution, the process movement adjustment and rescheduling methods are used to dynamically adjust the job scheduling plan. Based on the double-chain coding and greedy insertion method decoding rules, a hybrid variable neighborhood search NSGA-II algorithm is proposed. Aiming at the multiple constraints and high computational complexity of job shop scheduling problems, a compound heuristic rule based on machine selection is designed to obtain an initial population with uniform distribution, which is closer to the Pareto frontier, including the rules of minimum cumulative machine load and shortest processing time according to probability. In addition, the improved operation and equipment intersection strategy is adopted to improve the global search ability of the algorithm. Further, a variable neighborhood search strategy is designed based on several neighborhood structures selected by key processes and machines to further improve the local search ability of the algorithm. Through the numerical simulation of Kacem and Brandimarte standard instances and the statistical comparison with various representative algorithms, the effectiveness of the proposed algorithm is verified, and a better scheduling scheme is provided for the uncertain flexible job shop scheduling problem.

Keywords multi-objective optimization, flexible job shop scheduling, NSGA-II, variable neighborhood search, uncertain processing time, chance-constrained programming