

基于改进NSGA-III算法的电阻片车间多目标调度模型

姚鑫宇^{1,2}, 向忠^{1,2}

¹浙江理工大学机械工程学院, 浙江 杭州

²浙江理工大学平阳研究院有限公司, 浙江 温州

收稿日期: 2024年3月28日; 录用日期: 2024年5月3日; 发布日期: 2024年5月10日

摘要

氧化锌电阻片是轨道交通用避雷器的核心部件, 其生产制造过程面临规格型号多、工艺流程复杂, 人员换班频繁等问题。本研究以提高电阻片制造车间的订单交付率和能源利用率为目标, 提出了一种针对电阻片制造车间排产调度的基于多目标邻域的NSGA-III调度优化算法(DLNSGA-III)。首先, 构建了包含完工时间、总能耗和总负载的多目标柔性车间调度模型; 通过结合变邻域搜索算法、弱非支配排序和精英保留策略, 加强了算法的局部搜索能力、多样性和收敛性, 通过算例验证了所提算法的有效性和可行性。

关键词

氧化锌电阻片车间, NSGA-III, 多目标优化, 变邻域搜索算法, 弱非支配选择

Research on Flexible Scheduling Problem of Resistor Sheet Processing and Scheduling Based on Multi-Objective Neighborhood NSGA-III Algorithm

Xinyu Yao^{1,2}, Zhong Xiang^{1,2}

¹Faculty of Mechanical Engineering and Automation, Zhejiang Sci-Tech University, Hangzhou Zhejiang

²Zhejiang Sci-Tech University Pingyang Research Institute Co., Ltd, Wenzhou Zhejiang

Received: Mar. 28th, 2024; accepted: May. 3rd, 2024; published: May. 10th, 2024

Abstract

Zinc oxide resistors are the core component of lightning arresters for rail transit. The production

and manufacturing process faces problems such as multiple specifications and models, complex process flows, and frequent personnel changes. This study aims to improve the order delivery rate and energy utilization rate of the resistor sheet manufacturing workshop, and proposes a multi-objective neighborhood-based NSGA-III scheduling optimization algorithm (DLNSGA-III) for the production scheduling of the resistor sheet manufacturing workshop. First, a multi-objective flexible workshop scheduling model including completion time, total energy consumption and total load is constructed; by combining the variable neighborhood search algorithm, weak non-dominated sorting and elite retention strategy, the algorithm's local search capability, diversity and convergence, the effectiveness and feasibility of the proposed algorithm are verified through numerical examples.

Keywords

Zinc Oxide Resistor Sheet Workshop, NSGA-III, Multi-Objective Optimization, Variable Neighborhood Search Algorithm, Weak Non-Dominated Selection

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

电阻片制造包括混料、造粒、成型、排胶与烧结、绝缘层喷涂、磨片、铝电极喷涂、电性能检测等多道工序,是一种典型的多作业车间协作生产。在实际生产中,由于电阻片制造设备不便停机的特性,电阻片制造车间采用24小时轮班工作制,在车间复杂的工作环境下,班组的轮替,设备的突发故障,在订单逐渐增多的情况下,迟缓的处理过程对生产效率的影响越来越大,由此导致的能源利用率降低,订单延迟对企业的效益带来了巨大的负面影响。因此,根据实际的生产情况,及时制定合理的动态调度方案,减少负面事件对生产效率的影响,已经成为制造业和学术界关注的研究方向。

Achmad P. Rifai 等[1]通过应用改进的非支配排序生物地理优化算法(NSBBO)建立了双目标车间调度模型,提出了三种迁移模型(正弦、二次和梯形),并将非支配排序和拥挤距离排序特性引入到 NSBBO 中,通过对不同组合的作业数量、操作数量和机器数量的实例进行测试,研究评估了算法的性能,并分析了其效果、效率、鲁棒性和解的多样性,研究结果表明该模型是一种有效且有潜力的方法,可用于解决多目标优化问题。裴小兵等[2]提出了基于博弈论的改进 GA 算法的多目标柔性车间调度模型,通过将问题转化为动态博弈论模型,并应用改进混合遗传算法求解子博弈完美纳什均衡解,从而得到高质量的调度方案。实现了对多目标的快速优化部署。Guiliang Gong 等[3]以最小化最大完工时间和最小总能耗为目标,提出了一种分布式调度算法,研究采用了两个关键的改进方法:基于改进的初始化机制(BIM)和基于局部搜索优化(LSO)的改进,BIM 利用了模拟仿真来生成高质量的初始种群,以便更好地探索解空间。通过使用 BIM,可以更快地找到较优的解决方案,LSO 通过在遗传算法的迭代过程中应用局部搜索操作来改进当前种群的解决方案,加速了算法的收敛速度,并提高解的质量。Qianwang Deng 等[4]提出了以机器总负荷、制造工期和加工成本为目标的柔性车间调度模型,研究提出了一种新的蜂群进化引导策略,通过引入蜜蜂的觅食行为来指导算法的搜索过程。这种策略可以帮助算法更好地探索搜索空间,并提高算法的全局搜索能力。Himanshu Jain 等[5]基于 NSGA-III 算法,通过将 Deb 约束比较准则引入环境选择进行非支配排序提出了 C-NSGA-III,研究提出了一种基于参考点的非支配排序方法的进化多目标优化算

法, 并优化了约束条件, 并将其扩展为自适应方法。耿焕同等[6]提出了基于参考点的约束支配函数改进的 NSGA-III 算法, 用来提升 NSGA-III 算法的收敛性, 研究根据种群在决策空间的分布特征, 计算相邻两代种群的熵差, 判定种群的进化阶段。并且计算种群在目标空间的分布特征, 评估参考点的重要性。王俊艳等[7]提出了基于惩罚函数的进化算法, 该算法在基本 NSGA-II 算法的基础上引入了惩罚机制, 通过降低个体的优先级来减少其被选中的概率。欧阳洪才等[8]研究采用了两段式编码和基于参考点的小生境选择策略。提高了对多目标柔性作业车间求解效率, 通过两段式编码将染色体分为工序编码和机器编码两部分, 然后利用插入式贪婪解码将工序安排在最早可行的位置。通过基于参考点的小生境选择策略, 修剪合并的种群, 以保持种群的多样性和均匀分布。

NSGA-III 从 NSGA-II 的框架上发展而来, 针对 NSGA-II 中高维难以收敛及前沿解集均匀分布问题, NSGA-III 引入了参考点机制取代 NSGA-II 中的拥挤度机制, 从而获取了种群个体与响应参考点之间的映射关系, 进而使得种群更接近参考点的方向进化, 使得参考点的分布更加均匀。其中设定参考点数量为 H , 迭代 t 代的父代种群 P_t 和子代种群 Q_t 的种群数量均为 N , 将父代和子代的种群进行合并得 $U_t (U_t = P_t \cup Q_t)$, 在 U_t 中再选出 N 个体进入下代种群 P_{t+1} 。

本文以电阻片制造车间为对象, 创建了一个多目标柔性车间调度模型, 包含完工时间、总能耗和总负载。为了提高算法局部搜索能力引入了变邻域搜索算法。改进原有的非支配选择算法, 通过引入容忍参数实现弱非支配排序, 使其如果在其他目标上的性能明显更好, 就允许一个解在某个目标上的性能稍差, 从而提高了算法的多样性。

2. 问题描述及目标函数构建

2.1. 问题描述

在实际生产过程中, 电阻片制造车间会面临班组交替等常规干扰事件和机器故障等突发干扰时间, 从而导致生产效率降低, 生产成本提高。因此需要一个能够快速调整的生产调度计划能够在上述情况发生时快速生成最优方案, 保证生产的持续性和稳定性。

电阻片制造车间调度问题描述如下: 电阻片车间的工件总数为 n , 机器总数为 m , 集合描述为 $J = \{J_1, J_2, \dots, J_n\}$, $Z = \{Z_1, Z_2, \dots, Z_m\}$ 。对于每个工件 j_i 的工序集合为 $\{O_{i1}, O_{i2}, \dots, O_{ij}\}$, 每道工序都有至少一台以上的可选机器进行加工。参数模型如表 1 所示。

Table 1. Model parameters
表 1. 模型参数

符号	定义
n	工件总数
m	机器总数
i, h	工件号
J, K	工序号
z	机器号
q_i	工件 i 工序数
O_{ij}	工件 i 的第 j 道工序

续表

M_{ij}	工序 O_{ij} 的可选机器
S_{ij}	工序 O_{ij} 的开工时间
C_{ij}	工序 O_{ij} 的完工时间
$Load_z$	机器 z 负荷
$makespan_i$	工件 i 的完工时间
$makespan$	最大完工时间
$Load_all$	机器总负荷
E_all	总能耗
X_{ijz}	工序 O_{ij} 是否在机器 Z 上加工, 为 0~1 变量, 在 z 上加工为 1
G_{ijk}	工序 O_{ij} 和工序 O_{hk} 的先后顺序, 为 0~1 变量, ij 在前为 1
M	一个大正实数
T_{ijz}	工序 O_{ij} 在机器的加工时间
F_z	机器的 z 完工时间
P_z	机器的 z 负载功率
R_z	机器的 z 空载功率

2.2. 目标函数

在车间生产过程中, 不同的部门存在不同的考核指标, 这就需要模型具备多目标的处理能力, 从而最大限度的满足不同部门的实际需求。综合电阻片车间的实际需求, 在本模型中, 设置了三个目标函数分别是完工时间、总负载和总能耗。目标函数如下。

其中式(4)指开工后电阻片车间工序不能中断, 式(5)指任意电阻片车间工序在机器上只加工 1 次式(6)指任意电阻片车间工序的开工时间小于等于完工时间, 式(7)指任意工序完工时间小于等于最大完工时间, 式(8)指了每台机器在同一时刻只能加工一道电阻片车间工序, 式(9)指任意工件的上一道电阻片车间工序的完工时间不大于下一道电阻片车间工序的开工时间, 指任意电阻片车间工序的开工、加工、完工时间都非负。

1) 最小完工时间

$$makespan = \min \{ \max makespan_i \} \quad (1)$$

2) 总负载

$$Load_all = \sum Load_z \quad (2)$$

3) 总能耗

$$E_all = \sum Load_z P_z + (F_z - Load_z) R_z \quad (3)$$

多目标柔性作业车间调度数学模型为:

$$C_{ij} - S_{ij} = T_{ijz} \quad \forall i, j, z \quad (4)$$

$$\sum_{z=1}^n X_{ijz} = 1 \quad \forall i, j \quad (5)$$

$$S_{ij} + X_{ijz} T_{ijz} \leq C_{ij} \quad \forall i, j, z \quad (6)$$

$$C_{ij} \leq \text{makespan} \quad \forall i, j \quad (7)$$

$$X_{hks} C_{hk} + M(G_{hki} - 1) \leq X_{ijz} S_{ij} \quad \forall i, j, z, h, k \quad (8)$$

$$C_{ij} \leq S_{(j+1)} \quad \forall i, j \quad (9)$$

$$S_{ij} \geq 0, T_{ijz} \geq 0, C_{ij} \geq 0 \quad \forall i, j, z \quad (10)$$

3. 改进 NSGA-III 算法

Ded [9]提出的 NSGA-III 算法在多目标优化问题中有比较好的效果, 但仍有很多提升空间。

单一的种群初始化方式使得算法多样性大大降低, 会使算法更容易陷入局部最优解, DLNSGA-III 算法通过在种群初始化过程中引入混合初始化, 可以有效的提升算法多样性。

NSGA-III 算法中的交叉和变异操作可以让算法具有较强的全局搜索能力, 但同时也削弱了算法的局部搜索能力。DLNSGA-III 通过引入变邻域搜索算法, 提高了算法局部搜索能力。其中在车间调度优化的局部搜索研究中, 变邻域搜索算法得到了较为显著的提升[10]。为此, 根据氧化锌电阻片车间调度问题的特点, 本文采用了四种邻域结构: 1) 交换, 随机交换两个染色体解的位置; 2) 插入, 从染色体解中取出一个元素, 并将其插入到另一位置; 3) 反转逆序随机产生两个基因位置, 将两点间的片段反转; 4) 窗口式滑动, 选择一个由多个染色体构成的任务子集并将其向前或向后滑动一定数量的位置, 而不改变子集内的任务顺序。

在氧化锌电阻片多目标柔性车间调度过程中, 由于该车间调度的产品较为单一, 所以解的变化并不大, 本文设计了一种弱非支配排序替代原有的非支配排序, 对于小范围内变化的解提供了更广泛的解集。

在 NSGA-III 中, 直接选取子代种群为最优解集可能导致遗失迭代中的最优解。为了维护种群的多样性并确保收敛性, 引入了一个精英记忆库。

3.1. 算法流程

算法流程如图 1 所示。

DLNSGA-III 算法的具体步骤如下:

步骤 1: 参数设置: 令种群大小为 N ; 迭代总次数为 gen ; 最大迭代次数 $MAXGEN$; 交叉概率为 p_c ; 变异概率为 P_m 。

步骤 2: 初始化种群: 生成初始种群 P_t , 将迭代总次数设置为 0。

步骤 3: 根据优化目标及分段数, 生成均匀分布的参考点。

步骤 4: 当迭代总次数达到最大迭代次数时($gen > MAXGEN$), 则输出 Pareto 非劣解; 当迭代总次数小于最大迭代次数时($gen < MAXGEN$), 则记录当前最优解。

步骤 5: 对种群进行交叉和变异操作。

步骤 6: 完成交叉和变异操作后, 合并子代与父代个体。

步骤 7: 对合并后的种群进行变邻域搜索, 输出最优个体。

步骤 8: 用变邻域搜索的最优个体代替原种群的最差个体后, 对其使用弱非支配排序进行分层。

步骤 9: 对每一层进行基于参考点的小生境选择, 直到选取个数到 N , 转到步骤 4。

步骤 10: 若满足迭代要求, 则输出 Pareto 非劣解。否则继续步骤 5。

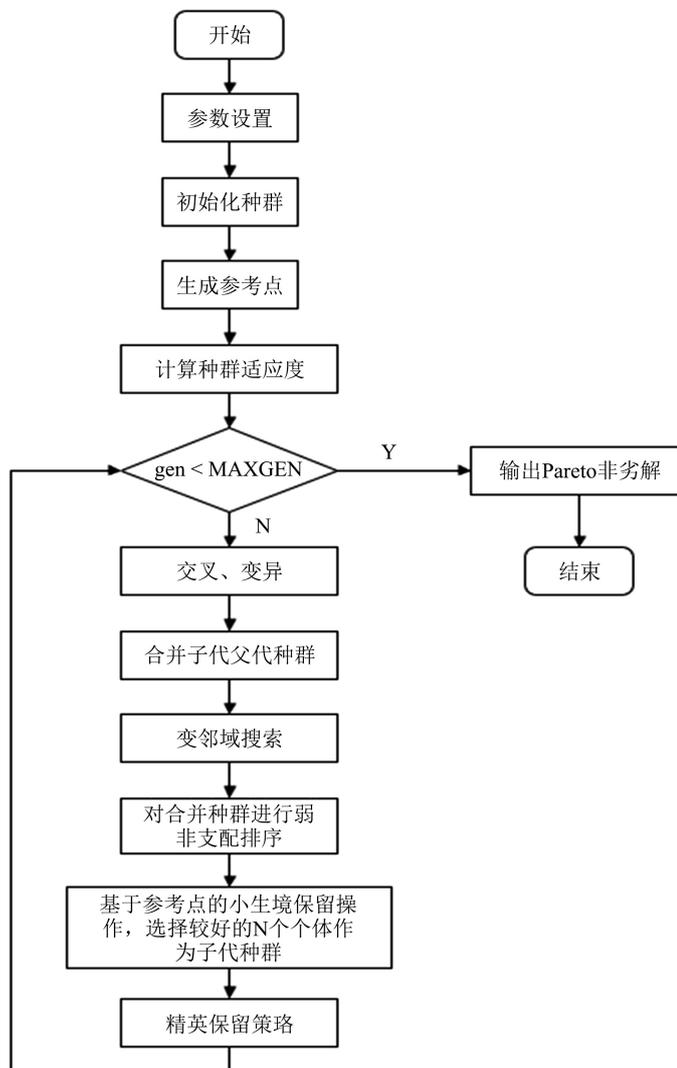


Figure 1. DLNAGA-III algorithm flow chart
图 1. DLNAGA-III 算法流程图

3.2. 混合初始化

种群的初始设置在 FJSP 中是关键, 因为初始解的优劣会影响到求解的速度和效果。其中, 机器的选择相较于工序的顺序更为核心。此处, 采用混合初始化方式来决策机器的选择, 具体包括全局选择(GS)、局部选择(LS)以及随机选择(RS) [11]。GS 的目标依次安排每个工件的加工, 每道工序选择最小负荷的机器。LS 依次安排每个工件的加工, 每次安排完一个工件加工后, 各个机器的负荷清 0, 每道工序选择最小负荷的机器。相对地, RS 依次安排每个工件的加工, 每道工序随机选择可加工机器, 这样可以确保调度解在目标空间中的分布更为分散, 同时也保证了种群的多样性。

3.3. 变邻域搜索算法

根据电阻片车间柔性调度问题, 采用了 4 种产生最优解邻域的方法。

两点交换: 随机交换两个染色体解的位置, 例如, 对于一个排列 1, 2, 3, 4, 5, 交换第 2 个和第 4 个元素的位置得到 1, 4, 3, 2, 5。

插入: 从染色体解中取出一个元素, 并将其插入到另一位置, 例如, 对于同样的排列 1, 2, 3, 4, 5, 将第 3 个元素取出并插入到第 5 个元素之后, 得到 1, 2, 4, 5, 3。

反转逆序: 选择序列中的两个位置 i 和 j , 其中 $i < j$, 然后反转从位置 i 到位置 j 的所有元素。例如, 对于序列: 1, 2, 3, 4, 5 如果选择位置 2 和 4, 则执行反转逆序操作后的序列为: 1, 4, 3, 2, 5。

窗口式滑动: 基本原理是在给定解的某个子序列或“窗口”上进行局部修改, 而不是对整个解进行全局修改。包括以下步骤:

窗口选择: 从当前解中随机选择一个连续的子序列, 也称为“窗口”。窗口的大小可以是预定义的, 或可以从一个给定的范围内随机选择。

滑动操作: 一旦选择了窗口, 就将这个子序列向前或向后“滑动”一定数量的位置。滑动的方向和距离可以是预定义的, 或可以随机决定。例如, 通过将上述的 3 个任务的窗口向前滑动 2 个位置。

调整其他任务: 滑动窗口后, 原来窗口的位置会空出来, 或者新的位置可能已经被其他任务占用。因此, 需要对那些被影响的任务进行相应的调整。最简单的方法是将被滑动窗口“推出”的任务移到原来窗口的位置, 以保持所有任务的连续性。

例如, 考虑一个简化的调度问题, 其中任务序列为: A, B, C, D, E, F。如果选择了一个包含 B 和 C 的窗口, 并将其向前滑动 1 个位置, 那么新的任务序列将是: B, C, A, D, E, F。

窗口式滑动的优点是它允许在解的局部区域进行精细的修改, 而不是进行全局的大范围修改。这有助于在搜索空间中进行更深入、更具针对性的探索。

基于上述 4 种邻域结构, 变邻域搜索算法流程如图 2 所示:

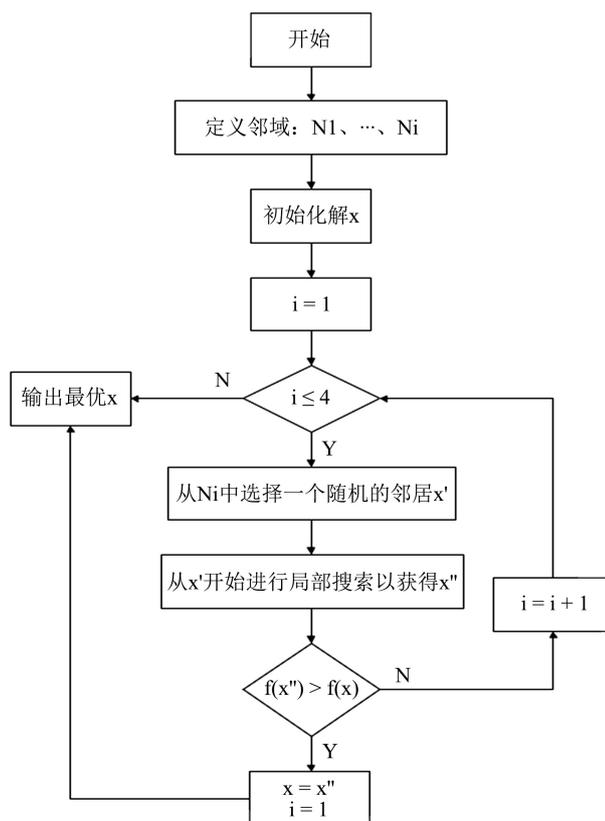


Figure 2. Variable neighborhood search algorithm flow chart

图 2. 变邻域搜索算法流程图

具体步骤如下:

步骤 1: 定义邻域结构, 设定邻域结构的集合为 $N_i \{i|1,2,3,4\}$ 。

步骤 2: 初始化, 选择一个初始解 x 。

步骤 3: 设置当前邻域索引 $i=1$, 并设定终止条件 $i \leq 4$ 。

步骤 4: 在当前邻域结构 N_i 中随机选择一个邻居 x' 。

步骤 5: 在 x' 为起始解的情况下进行局部搜索, 得到 x'' 。

步骤 6: 如果 x'' 比 x' 更好, 更新 $x=x''$ 并重新设置 $i=1$, 否则 $i+1$ 。

步骤 7: 输出最优解 x 。

本文考虑 NSGA-III 和 VNS 算法的各自优势以及电阻片车间柔性调度问题的特点, 提出了使用弱非支配排序的混合变邻域改进 NSGA-III。

3.4. 弱非支配排序

本文设计了一种弱非支配排序, 与传统的非支配排序相比, 弱非支配排序在处理解的等级分配时更为宽松, 因为它允许解在某些目标上与其他解相同, 而不被其他解支配

为了明确地描述这个关系, 设有两个解 x_1 和 x_2 , 并且有 m 个目标函数 f_1, f_2, \dots, f_m 。

对于每一个目标函数 f_i , 可以得到其在解 x_1 和 x_2 上的取值, 即 $f_i(x_1)$ 和 $f_i(x_2)$ 。

弱非支配关系可以定义如下:

解 x_1 弱支配解 x_2 当且仅当满足以下两个条件:

1) 对于所有的 $i \in \{1, 2, \dots, m\}$, 有 $f_i(x_1) \leq f_i(x_2)$;

2) 至少对于一个 $j \in \{1, 2, \dots, m\}$, 有 $f_j(x_1) < f_j(x_2)$ 。

根据这个定义, 如果 x_1 弱支配 x_2 , 那么 x_1 在所有目标上都不劣于 x_2 , 并且 x_1 在至少一个目标上优于 x_2 。

3.5. 精英保留策略

由于在迭代过程中会出现丢失最优解的问题, 于是在变邻域操作产生子代种群后, 建立了精英种群库[12], 种群库的容量为 N , 将子代种群与当前精英库合并, 并进行去重; 再利用弱非支配原则对其进行分层, 随后选取非支配层中的 F_1 存入精英库, 设记忆库的最大容量为 N 。在每一轮的迭代结束时, 都要更新这个记忆库。若当前非支配层 F_1 中的个体数少于或等于 N , 记忆库的容量 N_0 就被定义为 F_1 中的个体数和 N 中的较小值, 即 $N_0 = \min\{|F_1|, N\}$ 。

4. 实验例证

4.1. 实验构造及参数设置

DLNSGA-III 算法开发基于 Python 3.11.1, 主要依赖以下几个关键的第三方库, 包括 NumPy 1.24.3 和 Matplotlib 3.7.1。该算法在 Windows 11 操作系统上运行, 配备了 Intel i5-12600K 处理器和 16GB 的 RAM。在进行实验时, 采用了如下命名方式——j*c*a [13], 其中“*”代表一个正整数。例如, j10c6a1 代表的是一个涉及 10 个工件和 6 个加工步骤的案例, 数字 1 指的是它是第一个案例。这些数据是通过 Python 脚本随机生成的, 其中包括每台机器的加工时间(介于 15 到 40 分钟之间)、每个步骤涉及的机器数量(2 到 5 台之间)以及每台机器的能耗(5 到 10 kw 之间)。

为了测试算法的有效性, 随机生成了 20 个不同的案例, 并在以下几个特定场景中进行了测试: j10c5a1 至 j10c5a5、j20c6a1 至 j20c6a5、j30c7a1 至 j30c7a5、j50c8a1 至 j50c8a5。在这些案例中, 分别应用了 NSGA-II、

NSGA-III 以及 DLNSGA-III 算法进行求解, 并对这三种算法的性能进行了比较分析。这种比较旨在评估 DLNSGA-III 算法在处理不同规模和复杂性任务时的效率和准确性。

3 种算法参数设置为: 种群规模为 50, 迭代次数为 200, 交叉概率为 0.7, 交叉概率为 0.3。

4.1.1. 评价指标

欧式距离(Generational distance, GD) [14], 通过求取帕累托前沿上每个点到最近的真实帕累托上的欧式距离, 并取平均, 从而对算法的收敛性进行评价。

反转世代距离(Inverted generational distance, IGD) [14], 通过将真实帕累托前沿上每个点到算得的帕累托前沿上距离最近的点相加取平均, 从而对算法的多样性和收敛性进行评价。

非支配解(Non-dominated solution, NDS) [15], 通过统计非支配解的个数, 从而对算法的多样性进行评价。

4.1.2. 结果对比

表 2~4 展示了三种算法——DLNSGA-III、NSGA-II 和 NSGA-III——独立运行 10 次所得到的性能指标 GD、IGD 和 NDS 的结果。表 1 详细记录了 20 组实验中每个指标的最小值、最大值和平均值, 其中最优值以加粗形式标出。图 3~5 分别展示了三种算法在 GD、IGD 和 NDS 的平均值。图 6~8 则分别展示了这三种算法在 GD、IGD 和 NDS 的箱线图。

Table 2. GD

表 2. 欧式距离

案例	设备布局	NSGA-II			NSGA-III			DLNSGA-III		
		最小值	最大值	平均值	最小值	最大值	平均值	最小值	最大值	平均值
j10c5a1	2, 2, 5, 2, 2	0.0582	0.1112	0.0855	0.0427	0.0826	0.0623	0.0110	0.0378	0.0230
j10c5a2	3, 2, 2, 5, 3	0.0449	0.1321	0.0898	0.0234	0.0625	0.0437	0.0165	0.0376	0.0238
j10c5a3	2, 3, 5, 2, 2	0.0581	0.1372	0.0885	0.0242	0.0701	0.0533	0.0168	0.0362	0.0248
j10c5a4	3, 3, 2, 2, 5	0.0483	0.1285	0.0852	0.0238	0.0629	0.0446	0.0180	0.0361	0.0261
j10c5a5	5, 3, 2, 2, 2	0.0602	0.1308	0.0902	0.0423	0.0713	0.0699	0.0118	0.0361	0.0231
j20c6a1	2, 2, 5, 2, 2, 2	0.0725	0.1301	0.0950	0.0622	0.0965	0.0790	0.0134	0.0349	0.0208
j20c6a2	3, 3, 2, 2, 5, 2	0.0689	0.1341	0.0924	0.0613	0.1102	0.0752	0.0128	0.0377	0.0230
j20c6a3	5, 2, 2, 3, 4, 2	0.0662	0.1358	0.0973	0.0429	0.1034	0.0729	0.0148	0.0342	0.0218
j20c6a4	2, 3, 4, 2, 3, 3	0.0838	0.1332	0.1176	0.0511	0.1097	0.0851	0.0158	0.0301	0.0241
j20c6a5	3, 5, 4, 4, 2, 2	0.0822	0.1400	0.1096	0.6980	0.1174	0.0914	0.0101	0.0360	0.0224
j30c7a1	2, 2, 5, 2, 2, 2, 3	0.0812	0.1481	0.1220	0.0734	0.1223	0.1026	0.0111	0.0357	0.0248
j30c7a2	4, 2, 3, 3, 5, 2, 2	0.0839	0.1614	0.1152	0.0756	0.1350	0.1163	0.0112	0.0381	0.0240
j30c7a3	2, 3, 3, 2, 2, 4, 3	0.0981	0.1693	0.1271	0.0795	0.1387	0.1025	0.0121	0.0321	0.0220
j30c7a4	2, 5, 3, 3, 2, 2, 2	0.0955	0.1931	0.1478	0.0801	0.1541	0.1296	0.0110	0.0379	0.0242
j30c7a5	3, 2, 2, 2, 4, 3, 5	0.1175	0.2015	0.1661	0.0913	0.1662	0.1302	0.0120	0.0352	0.0241

续表

j50c7a1	2, 2, 5, 2, 2, 2, 3	0.1016	0.1945	0.1482	0.0816	0.1492	0.1298	0.0118	0.0345	0.0221
j50c7a2	4, 3, 2, 2, 5, 3, 4	0.1078	0.1986	0.1489	0.0898	0.1539	0.1375	0.0112	0.0324	0.0231
j50c7a3	5, 2, 2, 3, 3, 5, 3	0.1105	0.1992	0.1552	0.0926	0.1683	0.1236	0.0104	0.0345	0.0241
j50c7a4	3, 4, 2, 2, 5, 4, 4	0.1198	0.2101	0.1672	0.0958	0.1762	0.1466	0.0112	0.0311	0.0211
j50c7a5	2, 3, 2, 2, 4, 3, 3	0.1213	0.2110	0.1721	0.1098	0.1834	0.1501	0.0122	0.0337	0.0204
平均值		0.0840	0.1600	0.1210	0.0971	0.1217	0.0973	0.0128	0.0351	0.0231

Table 3. IGD
表 3. 反转世代距离

案例	设备布局	NSGA-II			NSGA-III			DLNSGA-III		
		最小值	最大值	平均值	最小值	最大值	平均值	最小值	最大值	平均值
j10c5a1	2, 2, 5, 2, 2	0.0172	0.0291	0.0226	0.0159	0.0262	0.0203	0.0072	0.0152	0.0175
j10c5a2	3, 2, 2, 5, 3	0.0159	0.0252	0.0191	0.0162	0.0248	0.0195	0.0083	0.0139	0.0100
j10c5a3	2, 3, 5, 2, 2	0.0181	0.0263	0.0205	0.0158	0.0237	0.0197	0.0073	0.0138	0.0101
j10c5a4	3, 3, 2, 2, 5	0.0184	0.0285	0.0214	0.0163	0.0256	0.0214	0.0071	0.0117	0.0098
j10c5a5	5, 3, 2, 2, 2	0.0184	0.0283	0.0215	0.0159	0.0226	0.0185	0.0070	0.0143	0.0102
j20c6a1	2, 2, 5, 2, 2, 2	0.0217	0.0293	0.0254	0.0162	0.0231	0.0198	0.0058	0.0099	0.0082
j20c6a2	3, 3, 2, 2, 5, 2	0.0189	0.0318	0.0257	0.0165	0.0258	0.0242	0.0068	0.0129	0.0093
j20c6a3	5, 2, 2, 3, 4, 2	0.0195	0.0254	0.0248	0.0171	0.0261	0.0221	0.0065	0.0158	0.0108
j20c6a4	2, 3, 4, 2, 3, 3	0.0192	0.0295	0.0261	0.0175	0.0266	0.0226	0.0056	0.0131	0.0092
j20c6a5	3, 5, 4, 4, 2, 2	0.0163	0.0294	0.0252	0.0177	0.0269	0.0230	0.0065	0.0097	0.0082
j30c7a1	2, 2, 5, 2, 2, 2, 3	0.0244	0.0318	0.0288	0.0182	0.0272	0.0241	0.0062	0.0131	0.0105
j30c7a2	4, 2, 3, 3, 5, 2, 2	0.0253	0.0319	0.0261	0.0184	0.0279	0.0231	0.0075	0.0131	0.0089
j30c7a3	2, 3, 3, 2, 2, 4, 3	0.0182	0.0311	0.0264	0.0179	0.0275	0.0218	0.0062	0.0162	0.0093
j30c7a4	2, 5, 3, 3, 2, 2, 2	0.0218	0.0332	0.0269	0.0171	0.0285	0.0222	0.0063	0.0108	0.0086
j30c7a5	3, 2, 2, 2, 4, 3, 5	0.0173	0.0324	0.0261	0.0165	0.0279	0.0234	0.0055	0.0136	0.0087
j50c7a1	2, 2, 5, 2, 2, 2, 3	0.0195	0.0331	0.0255	0.0172	0.0273	0.0228	0.0073	0.0136	0.0108
j50c7a2	4, 3, 2, 2, 5, 3, 4	0.0234	0.0319	0.0248	0.0175	0.0268	0.0237	0.0076	0.0140	0.0112
j50c7a3	5, 2, 2, 3, 3, 5, 3	0.0258	0.0335	0.0256	0.0182	0.0281	0.0241	0.0065	0.0139	0.0104
j50c7a4	3, 4, 2, 2, 5, 4, 4	0.0221	0.0341	0.0255	0.0165	0.0282	0.0239	0.0081	0.0142	0.0122
j50c7a5	2, 3, 2, 2, 4, 3, 3	0.0266	0.0349	0.0252	0.0186	0.0286	0.0243	0.0085	0.0147	0.0126
平均值		0.0204	0.0305	0.0247	0.0171	0.0265	0.0222	0.0069	0.0134	0.0103

Table 4. NDS
表 4. 非支配解

案例	设备布局	NSGA-II			NSGA-III			DLNSGA-III		
		最小值	最大值	平均值	最小值	最大值	平均值	最小值	最大值	平均值
j10c5a1	2, 2, 5, 2, 2	11	25	16.9	21	16.9	13	30	39	32
j10c5a2	3, 2, 2, 5, 3	13	24	19.2	16	26	25.5	28	40	33.5
j10c5a3	2, 3, 5, 2, 2	10	23	18	13	24	18.2	31	39	34.5
j10c5a4	3, 3, 2, 2, 5	15	29	19	15	24	18	30	39	33
j10c5a5	5, 3, 2, 2, 2	13	25	15.8	14	24	18.4	31	43	37.8
j20c6a1	2, 2, 5, 2, 2, 2	14	22	17.1	13	23	16.3	29	39	34
j20c6a2	3, 3, 2, 2, 5, 2	10	17	13	13	22	16.9	30	43	36
j20c6a3	5, 2, 2, 3, 4, 2	14	28	21	12	28	19.1	33	40	34.5
j20c6a4	2, 3, 4, 2, 3, 3	13	20	16.5	14	24	17.9	30	39	31.5
j20c6a5	3, 5, 4, 4, 2, 2	12	19	16.1	15	22	17.3	31	40	33.5
j30c7a1	2, 2, 5, 2, 2, 2, 3	10	21	16.6	15	21	16.5	29	39	33
j30c7a2	4, 2, 3, 3, 5, 2, 2	12	25	16.4	16	25	17.8	32	40	35.5
j30c7a3	2, 3, 3, 2, 2, 4, 3	17	28	21.1	16	25	21	33	42	36.6
j30c7a4	2, 5, 3, 3, 2, 2, 2	12	20	18.5	14	25	18.5	29	41	34.3
j30c7a5	3, 2, 2, 2, 4, 3, 5	16	30	22	14	24	19.8	25	36	32
j50c7a1	2, 2, 5, 2, 2, 2, 3	14	26	17.2	13	20	19	29	39	34
j50c7a2	4, 3, 2, 2, 5, 3, 4	16	29	19.7	15	21	19.5	34	43	36
j50c7a3	5, 2, 2, 3, 3, 5, 3	18	31	20.1	16	24	21.9	35	46	37.5
j50c7a4	3, 4, 2, 2, 5, 4, 4	20	33	21	18	27	23.5	35	47	38.3
j50c7a5	2, 3, 2, 2, 4, 3, 3	19	30	22.2	18	28	24	34	77.4	38
平均值		13.95	25.25	18.37	15.05	23.695	19.105	30.9	42.57	34.775

从图 3 的数据分析来看, 经过 10 次独立运行, DLNSGA-III 算法在 GD 指标上表现出显著优势, 其最小值、最大值和平均值的平均数分别为 0.0128、0.0351、0.0231, 这一结果明显优于 NSGA-II 算法的 0.0840、0.1600、0.1210, 以及 NSGA-III 算法的 0.0971、0.1217、0.0973。此外, 从图 4 的结果中可见, DLNSGA-III 在 IGD 指标上同样表现优异, 其最小值、最大值和平均值的平均数分别是 0.0069、0.0134、0.0103, 远优于 NSGA-II 的 0.0204、0.0305、0.0247 和 NSGA-III 的 0.0171、0.0265、0.0222。然而, 在图 5 中, DLNSGA-III 的 NDS 指标表现出色, 其最小值、最大值和平均值的平均数分别为 30.9、45.57、34.775, 显著高于 NSGA-II 的 13.95、25.25、18.37 和 NSGA-III 的 15.05、23.695、19.105。这一结果表明, DLNSGA-III 算法在收敛性和多样性方面优于其他两种算法。

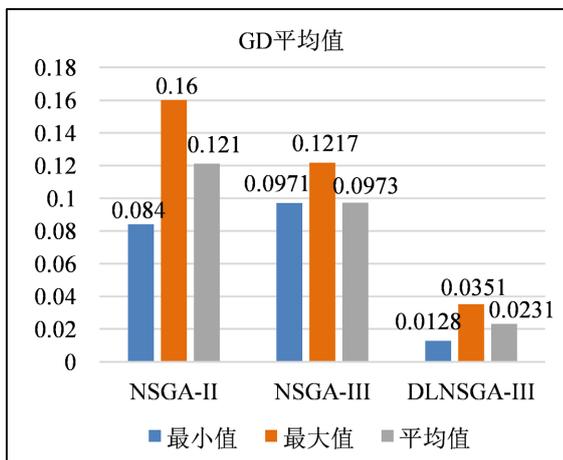


Figure 3. GD average
图 3. GD 平均值

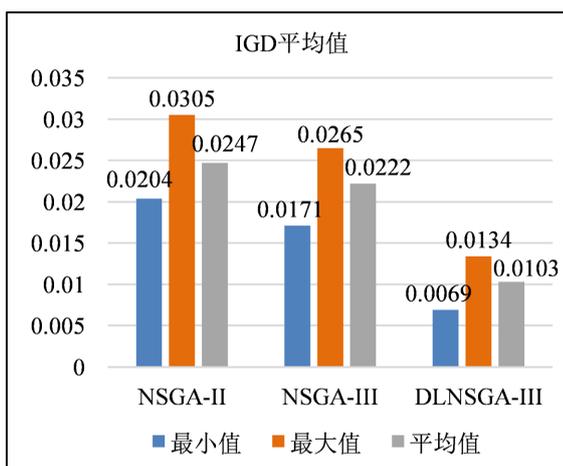


Figure 4. IGD average
图 4. IGD 平均值

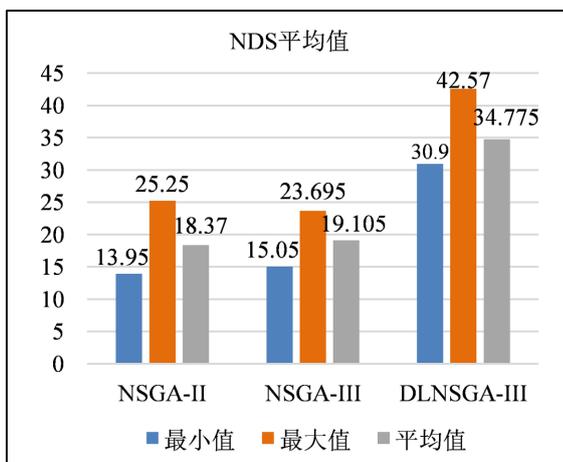


Figure 5. NDS average
图 5. NDS 平均值

通过对比表 2~4 中的 GD 值和 IGD 值, 可以看出随着案例数据规模的增大, 三种算法间的性能差距也逐渐扩大。例如, 在 j10c5a1 案例中, DLNSGA-III 的 GD 和 IGD 平均值分别为 0.0230 和 0.0175, 而 NSGA-II 和 NSGA-III 的对应的值分别为 0.0855 和 0.0226, 以及 0.0623 和 0.0203。在 j50c7a1 案例中, DLNSGA-III 的 GD 和 IGD 平均值分别为 0.0221 和 0.0175, 而 NSGA-II 和 NSGA-III 的对应的值分别为 0.1482 和 0.0255, 以及 0.1298 和 0.0228。这表明随着案例规模的增加, DLNSGA-III 在非劣解和非支配解集上的表现更加接近理想状态, 而 NSGA-II 和 NSGA-III 在这方面的差距则更加显著。

图 6 和图 7 显示, DLNSGA-III 的箱线图相比于其他两种算法更加扁平, 这表明其在各项指标上的方差更小, 具有更好的稳定性。而图 8 中显示的中位线更高, 表明 DLNSGA-III 在非劣解质量上更为优秀。

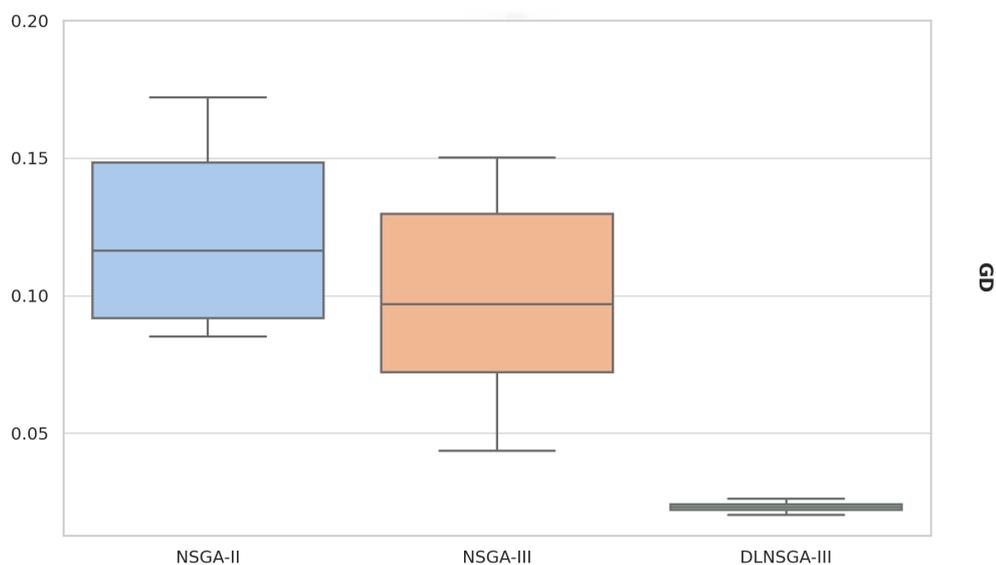


Figure 6. GD boxplot

图 6. GD 箱线图

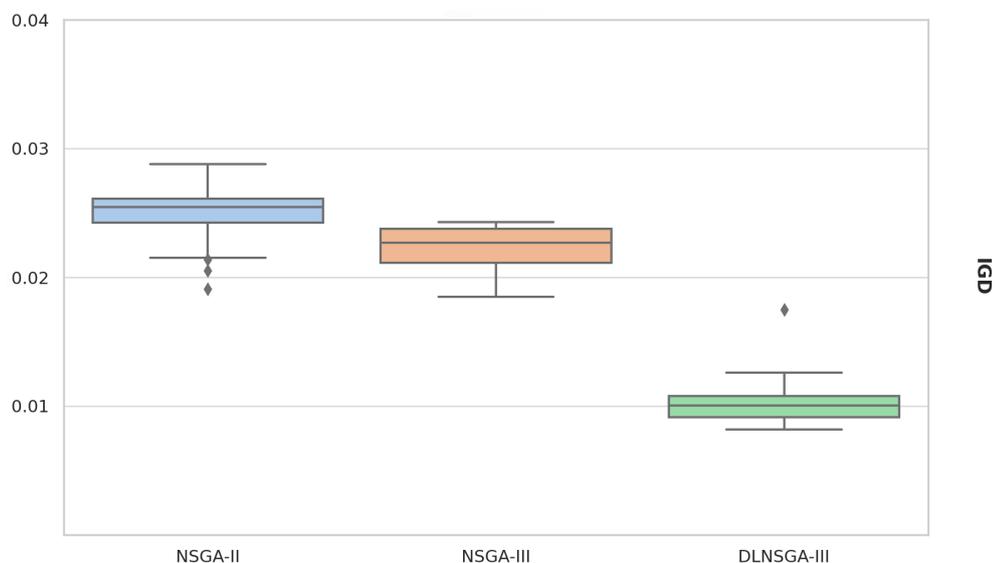


Figure 7. IGD boxplot

图 7. IGD 箱线图

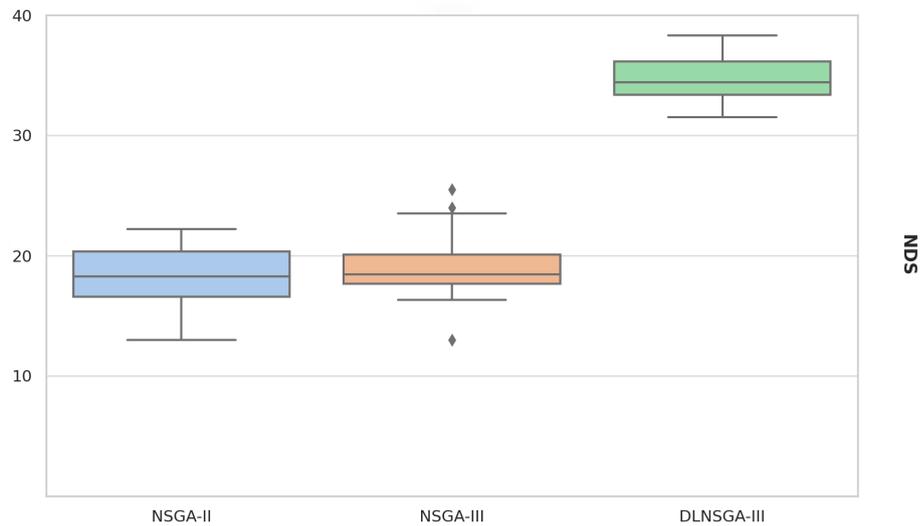


Figure 8. NDS boxplot
图 8. NDS 箱线图

4.2. 实例应用

某电气制造企业, 其生产的避雷器设备已经广泛用于高铁输电线。本文针对避雷器中氧化锌电阻片的生产; 氧化锌电阻片制造一般有 9 道工序: 混料, 造粒, 成型, 排胶与烧结, 绝缘层喷涂, 磨片, 铝电极喷涂, 电性能检测[16]。在车间的生产周期中, 需要在 25 台机器上生产 10 种电阻片订单。加工能耗, 加工时间, 工序可选择设备等实例数据详见表 5。其中混料机为 M1、M2, 造粒机为 M3、M4, 成型机为 M5、M6、M7、M8、M9, 排胶炉为 M10、M11, 烧结炉为 M12、M13, 侧面喷涂机为 M14、M15、M16, 绝缘层喷涂机为: M17、M18、M19, 磨片机为 M20、M21, 铝电极喷涂机为 M22、M23、M24, 电性能检测机 M25。

Table 5. Machine processing time and processing energy consumption
表 5. 机器加工时间和加工消耗

订单	加工时间(h)/机器负载(kw)												
	混料机		造粒机		成型机					排胶炉		烧结炉	
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13
J_1	15/22	15/24	20/55	21/50	25/53	25/53	25/53	24/45	24/45	15/650	15/600	20/850	21/800
J_2	16/22	13/22	23/57	20/48	24/52	24/52	24/52	25/46	25/46	10/650	14/600	22/850	23/800
J_3	11/23	13/22	22/56	24/53	24/52	24/52	24/52	26/46	26/46	14/650	15/600	24/850	23/800
J_4	15/24	16/25	23/57	24/53	25/53	25/53	25/53	22/43	22/43	11/650	10/600	24/850	26/800
J_5	16/22	18/26	24/58	25/55	27/55	27/55	27/55	28/48	28/48	13/650	15/600	26/850	28/800
J_6	18/26	16/25	24/58	25/55	24/52	24/52	24/52	27/47	27/47	16/650	17/600	22/850	25/800
J_7	19/27	20/27	21/55	23/52	23/52	23/52	23/52	24/45	24/45	15/650	15/600	25/850	24/800
J_8	22/29	18/26	25/58	22/51	25/53	25/53	25/53	22/43	22/43	17/650	17/600	26/850	28/800
J_9	20/27	20/27	25/58	22/51	24/53	24/53	24/53	26/47	26/47	18/650	20/600	27/850	28/800
J_{10}	21/27	20/27	24/58	22/51	29/56	29/56	29/56	28/48	28/48	21/650	24/600	21/850	25/800

续表

订单	加工时间(h)/机器负载(kw)											
	侧面喷涂机			绝缘层喷涂机			磨片机		铝电极喷涂机			检测机
	M14	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24	M25
J_1	11/35	11/35	10/30	20/35	25/30	20/35	32/40	32/40	20/30	22/36	22/36	34/35
J_2	12/35	12/35	11/30	21/35	24/30	21/35	31/40	31/40	22/21	23/36	23/36	32/33
J_3	13/35	13/35	12/30	22/35	24/30	22/35	32/40	32/40	19/30	21/36	21/36	35/35
J_4	13/35	16/35	13/31	23/36	25/31	23/36	33/42	33/42	18/30	22/36	22/36	35/35
J_5	16/37	16/37	14/31	24/36	27/36	24/36	34/42	34/42	18/30	25/38	25/38	31/33
J_6	14/35	14/35	15/32	25/37	24/30	25/37	35/42	35/42	23/21	26/38	26/38	32/33
J_7	15/36	15/36	16/32	26/37	23/29	26/37	36/44	36/44	25/23	27/38	27/38	36/35
J_8	16/37	16/37	17/35	27/37	25/30	27/37	33/42	33/42	24/23	29/39	29/39	31/33
J_9	18/38	18/38	18/35	28/38	24/30	28/38	34/42	34/42	24/23	26/38	26/38	40/38
J_{10}	19/38	19/38	18/35	27/37	29/38	27/37	32/40	32/40	26/24	25/38	25/38	43/39

由上述实验可知 DLNSGA-III 算法优于 NSGA-II 和 NSGA-III, 表 6 是三种算法的实例对比数据表, 从表中可知 DLNSGA-III 在完工时间、机器总负荷、总能耗三项指标的对比中均优于 NSGA-II 和 NSGA-III, 进一步验证了上述实验的结果。订单经过基于 DLNSGA-III 算法的排产优化后, 最大完工时间为 14.2 h, 机器总负荷为 3892 KW, 总能耗为 186 KW/h, 根据原本日产 8000 片电阻片的人工排产调度计划相比, 订单交付率和能源利用率获得大幅提升。图 9 为根据 DLNSGA-III 所得到的调度排产甘特图。

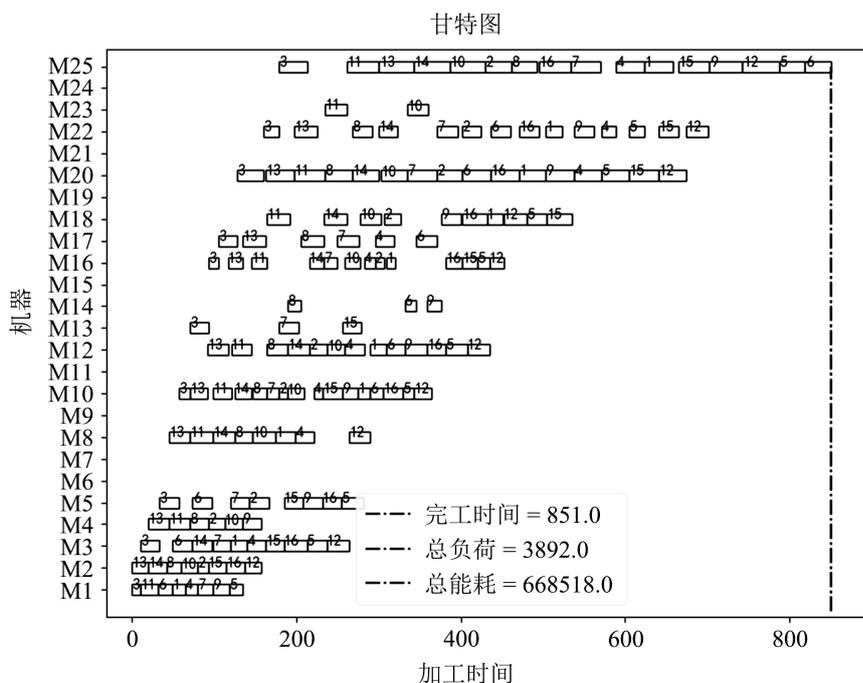


Figure 9. Scheduling plan gantt chart
图 9. 调度计划甘特图

Table 6. Example comparison experimental data table
表 6. 实例对比试验数据表

调度算法	完工时间/h	机器总负荷/kw	总能耗/kwh	迭代次数
NSGA-II	19.7	4382	224	200
NSGA-III	18.2	4017	203	200
DLNSGA-III	14.8	3885	186	200

5. 结论

建立以完工时间、总负载和总能耗为求解目标的问题模型, 提出 DLNSGA-II 算法求解此问题。

基于 NSGA-II 算法的缺陷, 结合变邻域搜索算法、弱非支配排序和精英保留策略, 加强了算法的局部搜索能力、多样性和收敛性。

研究选取了 20 组具有不同规模的案例, 采用 NSGA-II、NSGA-III 以及 DLNSGA-III 三种多目标优化算法进行求解。为了更直观、准确地对比这三种算法的性能, 绘制了详细的试验结果数据表格, 并通过箱线图形式展现了每种算法在各个案例中的表现。经过综合比较, 可以明确观察到 DLNSGA-III 在求解非支配解集方面的表现优于其他两种算法。实例数据对比也验证了 DLNSGA-III 在电阻片车间调度排产中优于 NSGA-II 和 NSGA-III。

从 DLNSGA-III 所得到的调度排产甘特图所呈现的结果说明, 与原本的人工调度排产方法相比, 基于 DLNSGA-III 算法的调度排产方法有效的提高了订单交付率和能源利用率。

参考文献

- [1] Rifai, A.P., Nguyen, H.T., Aoyama, H., *et al.* (2018) Non-Dominated Sorting Biogeography-Based Optimization for Bi-Objective Reentrant Flexible Manufacturing System Scheduling. *Applied Soft Computing*, **62**, 187-202. <https://doi.org/10.1016/j.asoc.2017.10.045>
- [2] 裴小兵, 李依臻. 基于三方博弈的改进遗传算法求解多目标柔性作业车间调度[J]. 工业工程与管理, 2020, 25(4): 59-68, 94.
- [3] Gong, G., Chiong, R., Deng, Q. and Luo, Q. (2020) A Memetic Algorithm for Multi-Objective Distributed Production Scheduling: Minimizing the Makespan and Total Energy Consumption. *Journal of Intelligent Manufacturing*, **31**, 1443-1466. <https://doi.org/10.1007/s10845-019-01521-9>
- [4] Deng, Q.W., Gong, G.L., Gong, X.R., *et al.* (2017) A Bee Evolutionary Guiding Nondominated Sorting Genetic Algorithm II for Multi-Objective Flexible Job-Shop Scheduling. *Computational Intelligence and Neuroscience*, **2017**, Article ID: 5232518. <https://doi.org/10.1155/2017/5232518>
- [5] Jain, H. and Deb, K. (2014) An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based-nondominated Sorting Approach, Part I: Handling Constraints and Extending to an Adaptive Approach. *IEEE Transactions on Evolutionary Computation*, **18**, 602-622. <https://doi.org/10.1109/TEVC.2013.2281534>
- [6] 耿焕同, 戴中斌, 王天雷, 等. 基于参考点选择策略的改进型 NSGA-III 算法[J]. 模式识别与人工智能, 2020, 33(3): 191-201.
- [7] 王俊艳. 惩罚策略辅助的快速非支配排序遗传算法 II 研究[J]. 重庆理工大学学报(自然科学), 2022, 36(6): 200-206.
- [8] 欧阳洪才, 张桐瑞, 吴定会. 基于改进 NSGA-III 算法的多目标柔性作业车间调度[J]. 控制工程, 2023, 30(1): 105-112.
- [9] Deb, K. and Jain, H. (2014) An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Transaction on Evolutionary Computation*, **18**, 577-601. <https://doi.org/10.1109/TEVC.2013.2281535>
- [10] 蔡玮, 赵轶, 陈浩杰, 等. 改进遗传变邻域算法求解飞机装配线调度问题[J]. 制造业自动化, 2021, 43(4): 69-73, 89.

-
- [11] Gao, L., Zhang, G.H. and Wang, X.J. (2012) Flexible Job Shop Scheduling Intelligent Algorithm and Its Application. Huazhong University of Science and Technology Press, Wuhan.
- [12] 罗凤鸣, 吕方林, 侯宗琰. 基于精英保留策略与爆炸算子的改进遗传算法[J]. 西华大学学报(自然科学版), 2018, 37(3): 83-88.
- [13] 王静云, 王雷, 蔡劲草, 等. 多目标混合流水车间调度问题求解算法[J]. 南京航空航天大学学报, 2023, 55(3): 544-552.
- [14] 慧荣, 李莉. 均匀设计表的 MATLAB 实现[J]. 统计与决策, 2008, 21(6): 144-146.
- [15] 李进, 李二超. 基于正态分布和自适应变异算子的 ϵ 截断算法[J]. 山东大学学报(工学版), 2019, 49(2): 47-53.
- [16] 王兰义, 任鑫, 黄海, 等. 国内外避雷器用氧化锌电阻片的技术现状与发展趋势[J]. 电瓷避雷器, 2021(6): 30-44.