

计算机集成制造系统
Computer Integrated Manufacturing Systems
ISSN 1006-5911, CN 11-5946/TP

《计算机集成制造系统》网络首发论文

题目：考虑准备时间的分布式混合零空闲置换流水车间调度问题
作者：陈水琳，郑建国
收稿日期：2023-07-05
网络首发日期：2023-09-27
引用格式：陈水琳，郑建国. 考虑准备时间的分布式混合零空闲置换流水车间调度问题[J/OL]. 计算机集成制造系统.
<https://link.cnki.net/urlid/11.5946.TP.20230927.1142.010>



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

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

考虑准备时间的分布式混合零空闲置换流水车间调度问题

陈水琳, 郑建国⁺

(东华大学 旭日工商管理学院, 上海 200051)

摘要: 分布式制造已成为经济全球化背景下的一种主流生产模式, 针对考虑序列相关准备时间的分布式混合零空闲置换流水车间调度问题 (Distributed mixed no-idle permutation flowshop scheduling problem with sequence-dependent setup times, DMNIPFSP/SDST), 以最小化最大完工时间、总能耗以及总拖期为目标, 构建了 DMNIPFSP/SDST 问题的数学模型, 根据问题的特征, 一种多目标灰狼优化算法 (Multi-objective grey wolf optimizer, MOGWO) 被提出。为了提高初始解的质量和多样性, 提出了三种改进的初始化策略以及随机生成相结合的初始化方法; 基于所求解问题的特点, 设计了一种离散化种群更新机制, 以平衡算法的全局探索和局部开发能力; 为了进一步提高解的质量和求解精度, 针对不同优化目标各提出了四种邻域策略进行局部搜索。仿真结果表明, 所提出的方法在解决多目标 DMNIPFSP/SDST 问题时具有优越的性能。

关键词: 分布式混合零空闲置换流水车间调度; 序列相关准备时间; 灰狼优化算法; 多目标优化

中图分类号: TP18

文献标志码: A

Distributed mixed no-idle permutation flowshop scheduling problem with setup times

CHEN Shuilin, ZHENG Jianguo⁺

(Glorious Sun School of Business and Management, Donghua University, Shanghai 200051, China)

Abstract: Distributed manufacturing has become a mainstream production mode in economic globalization. For the distributed mixed no-idle permutation flowshop scheduling problem with sequence-dependent setup times (DMNIPFSP/SDST), with the objective of minimizing the maximum makespan, total energy consumption and total tardiness, a mathematical model of the DMNIPFSP/SDST problem is constructed. According to the characteristics of the problem, a multi-objective grey wolf optimizer (MOGWO) is proposed. To improve the quality and diversity of the initial solution, three improved initialization

收稿日期: 2023-07-05; 修订日期: 2023-09-19. Received 05 July 2023; accepted 19 Sep. 2023.

基金项目: 中央高校基本科研业务费专项资金、东华大学研究生创新基金资助 (CUSF-DH-D-2023053)。

Foundation item: Project supported by the Fundamental Research Funds for the Central Universities and Graduate Student Innovation Fund of Donghua University, China (No. CUSF-DH-D-2023053).

strategies and the random generation method are proposed. Based on the characteristics of the problem to be solved, a discrete population update mechanism is designed to balance the global exploration and local exploitation. To further improve the quality and accuracy of the solution, four neighborhood strategies are proposed for local search for different optimization objectives. Simulation results show that the proposed method can achieve superior performance in solving the multi-objective DMNIPFSP/SDST.

Keywords: distributed mixed no-idle permutation flowshop scheduling; sequence-dependent setup times; grey wolf optimizer; multi-objective optimization

1 引言

在智能制造过程中,合理配置生产资源以获得更高的经济效益和客户满意度是至关重要的环节,因此,车间调度问题作为一个与生产调度密切相关的问题,受到广泛的关注^[1]。在车间调度中,分布式制造提高了产品质量,降低了生产成本,并使管理风险降到最低^[2]。为了适应多工厂生产环境,NADERI 和 RUIZ^[3]提出了分布式车间调度问题的定义。然而,零空闲置换流水车间调度问题(No-idle permutation flowshop scheduling problem, NIPFSP)是置换流水车间调度问题的推广,是其中非常重要的一类。在 NIPFSP 中,机器不允许处于空闲状态,即机器在加工完一个工件后应立即加工下一个工件,其主要广泛存在于玻璃加工、铸造厂、集成电路生产和炼钢等现代工业的组合优化问题。在实际生产过程中,往往会出现两种情况:第一,所有机器都属于零空闲状态是不太现实的,大多数机器允许空闲时间,而部分机器因成本高需要受到零空闲状态的约束,通常,将其称为混合零空闲置换流水车间调度问题(Mixed no-idle permutation flowshop scheduling problem, MNIPFSP);第二,机器需要在连续工件加工之间进行额外的操作,如机器清洗、刀具更换、工件运输等^[4],因此,还应考虑序列相关准备时间(Sequence-dependent setup times, SDST)。从现有文献来看,ALMEIDA 等^[5]提出了四种算法求解目标函数为最大完工时间最小化的序列相关准备时间零等待流水车间调度问题,结果表明,迭代贪婪算法求解效果良好。ABREU 和 NAGANO^[6]针对序列相关准备时间的开放式流水车间调度问题,设计了一种自适应大邻域搜索算法,仿真结果表明,所提出的方法在解决大规模算例具有一定的优势。GUO 等^[7]引入差分飞行策略改进果蝇优化算法,提出了一种离散的果蝇优化算法求解序列相关准备时间的分布式流水车间调度问题,研究表明,所提出的算法表现良好。MIYATA 和 NAGANO^[8]采用变邻域迭代贪婪算法求解考虑序列相关准备时间的分布式逆车间调度问题,通过在小规模和大规模算例验证所提出算法的性能,结果表明,所提出的算法优于已有文献

中的其他算法。KARABULUT 等^[9]通过使用自适应的进化算法求解以最小化最大完工时间为目标的序列相关准备时间的分布式流水车间调度问题, 研究结果表明, 所提出的算法具有较强的鲁棒性, 并且能在较短的计算时间内提供高质量的解决方案。从上述文献可看出, 第一, 大多只是从生产效率的角度对经济指标进行研究, 考虑能源和环境因素、客户满意度的研究相对较少; 第二, 搜索策略比较单一, 大多使用随机方式生成初始解, 好的初始化策略有利于提高种群的质量和多样性; 第三, 大多提出的算法仅适用于求解单目标调度优化问题, 而多目标优化问题解空间复杂, 需要进一步开发适用于求解多目标优化问题的算法。

近年来, 碳排放和能源消耗引起了广泛关注, 由于制造业是最大的能源消耗行业, 所以能源消耗往往是不可忽视的一部分。如果不控制能源消耗, 过度消耗最终将对自然环境和人类社会产生负面影响。因此, 节能调度在绿色制造系统中起着至关重要的作用, 主要归源于通过生产调度实现节能不需要过多的资本投入, 并且可以很容易地应用于现有的制造和生产环境中^[10]。除此之外, 在当今以顾客为导向的竞争市场中, 总拖期对一个企业的生存和发展也是至关重要^[11]。总拖期的最小化可以使产品尽快交付, 从而提高客户满意度, 减少企业损失。综上, 在本文中, 我们的优化目标不仅考虑 SDST 约束的分布式混合零空闲置换流水车间调度问题 (DMNIPFSP/SDST) 的最大完工时间最小化, 还考虑了总能耗和总拖期最小化。由于考虑单一目标的分布式混合零空闲置换流水车间调度问题 (Distributed mixed no-idle permutation flowshop scheduling problem, DMNIPFSP) 已经被证明是 NP-hard (Non-deterministic polynomial-hard) 问题^[12], 因此, 以最小化最大完工时间、总能耗和总拖期为优化目标的 DMNIPFSP/SDST 也是 NP-hard 问题, 具有更复杂的求解难度。当然, 随着问题规模的增大, 不太适合使用精确算法进行求解。此外, 在多目标的 DMNIPFSP/SDST 中需要解决两个子问题, 即为每个工件分配工厂和确定每个工厂中的工件顺序。由于这两个子问题是耦合的, 且不能按顺序求解, 所以提出有效的元启发式算法来求解多目标 DMNIPFSP/SDST 问题具有重要意义。

灰狼优化算法 (Grey wolf optimizer, GWO) 是 MIRJALILI 等^[13]提出的元启发式算法, 其灵感来源于灰狼捕食猎物活动的启发而开发的一种优化方法, 该算法具有简单的搜索机制以及快速收敛的特点, 并且其在解决复杂的优化问题方面表现出了强大的优势^[14-20]。LU 等^[14]引入了变邻域搜索策略改进灰狼优化算法的全局探索能力和局部开发能力, 并将其求解混合流水车间调度问题, 结果表明, 所提出的算法在求解这个问题上明显优于其竞争算法。雷斌等^[15]设计了一种改进的灰狼算法解决柔性车间调度问题, 并且引入了交叉变异策略增强算法的搜索能力。唐红涛等^[16]通过引入非线性

性函数和权重更新机制,并将所提出的算法求解模糊车间调度问题,仿真实验证明了所提算法的有效性。顾九春等^[17]针对车间的节能调度问题,采用多目标灰狼优化算法,通过排序规则产生初始种群,使用记忆池机制和邻域结构对个体进行位置更新,以 40 个生产车间调度问题为实例,对所提出的方法进行验证。LI 等^[18]提出了一种改进型灰狼优化算法求解分布式柔性车间调度问题,引入了四种交叉算子扩展搜索空间,通过与六种先进算法对比,发现所提出的算法具有良好的性能。WANG 等^[19]针对焊接车间逆调度问题,引入了多种邻域结构丰富灰狼优化算法的局部开发能力,通过三种实例验证所提出算法的性能,发现所提出的算法在求解逆调度问题具有一定的优越性。尽管诸多学者通过引入不同的策略对灰狼优化算法进行了改进,但对 GWO 的离散化以及多目标优化研究还比较少见,且很少将其解决以最小化最大完工时间、总能耗和总拖期为优化目标的 DMNIPFSP/SDST。因此,迫切需要提出一种有效的改进灰狼优化算法求解多目标 DMNIPFSP/SDST。

基于以上分析,本文提出一种多目标灰狼优化算法(Multi-objective grey wolf optimizer, MOGWO)求解考虑序列相关准备时间的分布式混合零空闲置换流水车间调度问题(DMNIPFSP/SDST)。首先,构建了一个以最小化最大完工时间、总能耗以及总拖期为目标数学模型;为了提高初始解的质量和多样性,提出了三种改进初始化策略以及随机生成相结合的初始化方法;其次,针对所求解问题的特点,提出了一种离散化的种群更新机制,以平衡算法的全局探索和局部开发能力;为了进一步提高解的质量和求解精度,针对不同的优化目标各提出了四种邻域策略进行局部搜索;最后,采用非参数方法进行统计对比分析。为了验证所提出算法的性能,通过对四种不同序列相关准备时间且不同规模的算例进行检验,并将其与多种算法进行对比,证明了 MOGWO 在求解 DMNIPFSP/SDST 具有一定的优势。

2 问题描述及数学模型

2.1 问题描述

本文探究了一种以最小化最大完工时间、总能耗以及总拖期为优化目标的序列相关准备时间的分布式混合零空闲置换流水车间调度问题(DMNIPFSP/SDST),该问题是在分布式混合零空闲置换流水车间调度问题(DMNIPFSP)的基础上进行扩展而来,主要解决两个子问题:①为每个工件分配合适的工厂;②确定每个工厂中的工件加工顺序。具体而言,车间有 n 个工件在 F 个工厂里进行加工,所有工厂都有相同的 m 台机器,每台机器的加工速度不完全相同,每个工件都能在任意的工厂进行加工。若一个工件在机器上高速加工,那么所需要的加工时间会更短,加工能耗会增加。在这种情况下,最大完工时间可能会减少,但是,总能耗会相应增加^[21],从而说明最大完工时间和总

能耗两个优化目标是相互冲突的，无法直接应用简单的单目标求解方法，使得求解方法具有一定的挑战性。值得一提的是，考虑单一目标的 DMNIPFSP 问题已经被证明是 NP-hard 问题^[12]，求解具有一定的难度，然而，本文不仅考虑了序列相关准备时间约束，还讨论了最大完工时间、总能耗以及总拖期的多目标优化，其中，总能耗包括加工能耗、序列相关准备时间能耗以及空闲能耗，因此，具有更复杂的求解难度。此外，考虑序列相关准备时间的分布式混合零空闲置换流水车间调度问题需要满足以下基本假设：

- ①不同机器上工件的加工顺序按同一次序进行。
- ②每个工件都要在指定的工厂中进行加工，在加工过程中，不能转移到另一个工厂。
- ③同一时间只可在一台机器上进行一个工件的加工，且每个工件一旦开始就不会中断。

2.2 符号定义

本文中的相关符号定义，详见表 1 所示。

表 1 符号定义

符号	说明
n	工件的数量
F	工厂的数量
m	机器的数量
k	工件在机器上加工的位置索引号
J_j	工件集合 $J = \{J_1, \dots, J_n\}$
M_i	机器集合 $M = \{M_1, \dots, M_m\}$
F_f	工厂集合 $F = \{F_1, \dots, F_F\}$
M'	含有零空闲约束的机器集合，如果不在 M' 集合中的机器定义为常规机器，允许工件之间有空闲
v_i	在机器 i 上的加工速度
$P_{i,j}$	工件 J_j 在机器 M_i 上的加工时间
$C_{f,i,j}$	在工厂 f 中，工件 J_j 在机器 M_i 上的完工时间
C_{\max}	所有工厂中的最大完工时间
$s_{j,l}^i$	在机器 M_i 上，工件 J_j 与 J_l 之间的序列相关准备时间 ($l \leq n, l \neq j$)
PP_i	在机器 M_i 上单位时间的加工能耗

SP_i 在机器 M_i 上单位时间的序列相关准备时间能耗

IP_i 在机器 M_i 上单位时间的空闲能耗

2.3 数学模型

基于以上的假设和相关符号定义，建立以最小化最大完工时间、总能耗以及总拖期为多目标函数的序列相关准备时间的分布式混合零空闲置换流水车间调度问题的数学模型，具体如下所示。

目标函数：

$$F_1 = \min \{C_{\max}\}$$

$$F_2 = \min \{PEC + SEC + IEC\}$$

$$F_3 = \min \left\{ \sum_{f=1}^F \sum_{j=1}^n \max(C_{m,j} - d_j, 0) \right\}$$

约束条件：

$$C_{\max} \geq C_{f,j} \quad \forall f \in \{1, \dots, F\}, \forall i \in \{1, \dots, m\}, \forall j \in \{1, \dots, n\} \quad (1)$$

$$\sum_{k=1}^n \sum_{f=1}^F X_{f,j,k} = 1, \forall j \in \{1, \dots, n\} \quad (2)$$

$$\sum_{j=1}^n \sum_{k=1}^n X_{f,j,k} = 1, \forall f \in \{1, \dots, F\} \quad (3)$$

$$X_{f,j,k} = \sum_{l=1}^n Y_{f,j,k,l} \quad \forall f \in \{1, \dots, F\}, \forall j \in \{1, \dots, n\}, \forall k \in \{2, \dots, n\} \quad (4)$$

$$X_{f,j,k} = \sum_{l=1}^n Y_{f,j,k,l} \quad \forall f \in \{1, \dots, F\}, \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, n-1\} \quad (5)$$

$$Y_{f,j,l,j} = 0 \quad \forall f \in \{1, \dots, F\}, \forall j, l \in \{1, \dots, n\} \quad (6)$$

$$C_{f,j,k} \geq \sum_{j=1}^n X_{f,j,k} \cdot \frac{P_{i,j}}{v_i} \quad \forall f \in \{1, \dots, F\}, \forall k \in \{1, \dots, n\} \quad (7)$$

$$C_{f,j,k} \geq C_{f,j-1,k} + \sum_{j=1}^n X_{f,j,k} \cdot \frac{P_{i,j}}{v_i} \quad \forall f \in \{1, \dots, F\}, \forall i \in \{2, \dots, m\}, \forall k \in \{1, \dots, n\} \quad (8)$$

$$C_{f,j,k} = C_{f,j,k-1} + \sum_{j=1}^n X_{f,j,k} \cdot \frac{P_{i,j}}{v_i} \quad \forall f \in \{1, \dots, F\}, \forall i \in M', \forall k \in \{2, \dots, n\} \quad (9)$$

$$C_{f,j,k} \geq C_{f,j,k-1} + \sum_{j=1}^n X_{f,j,k} \cdot \frac{P_{i,j}}{v_i} + \sum_{j=1}^n \sum_{l=1}^n Y_{f,j,k,l} \cdot s_{j,l}^i \quad \forall i \notin M', \forall f \in \{1, \dots, F\}, \forall k \in \{2, \dots, n\} \quad (10)$$

$$PEC = \sum_{f=1}^F \sum_{j=1}^n \sum_{i=1}^m PP_i \cdot X_{f,j,k} \cdot \frac{P_{i,j}}{v_i} \quad \forall k \in \{1, \dots, n\} \quad (11)$$

$$SEC = \sum_{f=1}^F \sum_{j=1}^n \sum_{l=1}^n \sum_{i=1}^m SP_i \cdot X_{f,j,k} \cdot X_{f,j,k+1} \cdot s_{j,l}^i \quad \forall k \in \{1, \dots, n\} \quad (12)$$

$$IEC = \sum_{f=1}^F \sum_{j=1}^n \sum_{i=1}^m IP_i \cdot X_{f,j,k+1} \cdot (C_{f,i,k+1} - C_{f,i,k} - s_{k,k+1}^i - \frac{P_{i,j}}{v_i}) \forall k \in \{1, \dots, n-1\} \quad (13)$$

$$C_{f,i,k} \geq 0 \forall f \in \{1, \dots, F\}, \forall i \in \{1, \dots, m\}, \forall k \in \{1, \dots, n\} \quad (14)$$

$$X_{f,i,k} \in \{0, 1\} \forall f \in \{1, \dots, F\}, \forall i \in \{1, \dots, m\}, \forall k \in \{1, \dots, n\} \quad (15)$$

$$Y_{f,i,k,j} \in \{0, 1\} \forall f \in \{1, \dots, F\}, \forall j, k, l \in \{1, \dots, n\} \quad (16)$$

F_1 表示最小化最大完工时间； F_2 表示最小化总能耗； F_3 表示最小化总拖期；约束（1）确保 C_{\max} 是所有工厂中的最大完工时间；约束（2）表示每个工件仅能分配到一个工厂加工；约束（3）表示每个位置最多放置一个工件；约束（4）确保在工厂 f 中 k 位置上工件 J_j 之前只有一个工件；约束（5）确保在工厂 f 中 k 位置上工件 J_j 之后只有一个工件；约束（6）表示排列在每个工厂的第一个位置的任意工件均没有序列相关准备时间；约束（7）和约束（8）确保工件只有在上一台机器加工完成才能进行下一台机器加工；约束（9）表示满足具有零空闲约束的机器的完工时间；约束（10）表示常规机器的完工时间；约束（11）表示加工状态下的能耗；约束（12）表示序列相关准备时间状态下的能耗；约束（13）表示空闲状态下的能耗；约束（14）表示每一个工件的完工时间是非负的；约束（15）和（16）定义了决策变量的取值。

3 多目标灰狼优化算法

3.1 编码与解码

由于 DMNIPFSP/SDST 中需要解决两个子问题：①每个工件分配工厂；②确定每个工厂中的工件加工顺序。基于此，本文采用 PAN 等^[22]提出的工件编码方式，其中，每个个体表示在可行域中的一个解，每个可行解 π 是由一个二维数组表示，共有 F 维，每一维表示一个工厂的工件排序，通过工件索引号代表该工厂的工件加工顺序。基于以上对编码的描述，一个可行解 π 可以直接表示为 F 个工厂所包含的工件排列集合，那么 $\pi = \{\pi_1, \pi_2, \dots, \pi_F\}$ ，其中， $\pi_i = \{\pi_{i,1}, \pi_{i,2}, \dots, \pi_{i,n_i}\}$ 表示该工厂中工件的加工顺序，并且 $\sum_{i=1}^F n_i = n$ ，即编码序列的总长度等于所有工件的数量。由于可行解 π 中既包含了每个工厂分配的工件也包含了工件在该工厂中的加工顺序，因此，该编码方式可以很容易地解码为一个 DMNIPFSP/SDST 的调度解。为了更好理解该编码和解码过程，本文给出一个包含 9 个工件（ $n=9$ ）和 2 个工厂（ $F=2$ ）的示例，假设该问题的一个可行解编码为 $\pi = \{\pi_1, \pi_2\}$ ，其中，

$\pi_1 = \{J_7, J_8, J_2, J_6, J_3\}$, $\pi_2 = \{J_5, J_4, J_1, J_9\}$, 具体如图 1 所示。那么对应的解码如下: 在工厂 1 中分配 J_2 ,

F_1	7	8	2	6	3
F_2	5	4	1	9	

J_3 , J_6 , J_7 以及 J_8 五个工件, 并按照 $J_7 \rightarrow J_8 \rightarrow J_2 \rightarrow J_6 \rightarrow J_3$ 的顺序进行加工; 在工厂 2 中分配 J_1 , J_4 , J_5 以及 J_9 四个工件, 并按照 $J_5 \rightarrow J_4 \rightarrow J_1 \rightarrow J_9$ 的顺序进行加工。

图 1 编码示意图

3.2 种群初始化

由于初始化方法会直接影响算法的求解效果, 因此, 通过多种策略的初始化方法可以有效地改善初始解质量以及增加种群的多样性。众所周知, LR (Liu-Reeves) 启发式算法在求解置换流水车间调度问题时表现出优异的性能^[23]。在此基础上, 根据 DMNIPFSP/SDST 的特征, 本文扩展了相关的索引函数, 提出了两种改进的 LR 规则生成初始种群, 分别表示为 LR1 和 LR2。具体的计算步骤如下所示:

改进的 LR1 规则的详细描述如下: ①计算每个工件在所有机器上完工时间的平均值、标准差以及偏度, 并将其进行求和; ②对工件 $j=1, 2, \dots, n$, 按照步骤①所计算出的结果以降序的顺序排列; ③将步骤②所获得的工件排序的前 F 个工件分别依次插入到 F 个工厂中; ④根据索引函数 (具体如公式 (17) - (19) 所示) 分别计算第 $F+1$ 个工件以及之后的工件的索引值; ⑤找到步骤④中所获得的最小索引值对应的工件; ⑥将步骤⑤获得的工件插入到当前完工时间最小的工厂后面; ⑦重复执行步骤④-步骤⑥, 直到分配完所有工件为止。

假设以工件 J_j 为例, 找到当前完工时间最小的工厂 g^* , 计算公式如下:

$$IF_{j,n_{g^*}} = (c - n_{g^*}) \times IT_{j,n_{g^*}} + C_{m,j} \quad (17)$$

$$IT_{j,n_{g^*}} = c \times m \times \sum_{i=2}^m \frac{t_{ij}}{c \times i + n_{g^*} \times (m - i)} \quad (18)$$

$$t_{ij} = \max \left\{ (C_{i-1,j} - C_{i[n_{g^*}]}), 0 \right\} \quad (19)$$

其中, c 是一个常数 ($c = n/F - 2$, n 表示工件数量, F 表示工厂数量); m 表示机器数量; n_g 表示工厂 g^* 中当前的工件数; $[n_g]$ 表示位置为 n_g 所对应的工件; $C_{m,j}$ 表示工件 j 在第 m 台机器上的完工时间。

改进的 LR2 规则的详细描述如下: ①计算每个工件在所有机器上完工时间; ②对工件 $j=1, 2, \dots, n$, 按照步骤①所计算出的结果以降序的顺序排列; ③将步骤②所获得的工件排序的前 F 个工件分别依次插入到 F 个工厂中; ④根据索引函数 (具体如公式 (17) - (19) 所示) 分别计算第 $F+1$ 个工件以及之后的工件的索引值; ⑤找到步骤④中所获得的最小索引值对应的工件; ⑥将步骤⑤获得的工件插入到当前完工时间最小的工厂后面; ⑦重复执行步骤④-步骤⑥, 直到分配完所有工件为止。

另外, 本文也提出了一种改进的 IFRB4 (Improved Farahmand-Ruiz-Borojerdian4) 规则生成初始种群, 具体的详细描述如下: ①计算每个工件在所有机器上完工时间的平均值、标准差以及偏度, 并将其进行求和; ②对工件 $j=1, 2, \dots, n$, 按照步骤①所计算出的结果以降序的顺序排列; ③选择排列中第 1 个工件, 再将第 2 个工件插入到第 1 个工件的全部可能位置, 最后选择一个完工时间最短的顺序; ④将第 3 个工件插入步骤③所选取的最好的排序的所有可能位置, 同理, 最终选取完工时间最小的排序; ⑤在步骤④所选取的最优位置上, 抽取其前后各两个工件, 将其重新插回所有位置, 若所得到的解优于步骤④所得到的解, 那么执行步骤⑥, 否则, 保持步骤④所获得的解决方案; ⑥同理, 按照步骤④的操作, 重复执行, 直到所有工件都被插入到上一个步骤的最优工件排序中, 方可结束以上操作。按照以上步骤所获得的工件序列进行分配工厂, 假设将 n 个工件分配到 F 个工厂, 具体步骤如下: ①将改进的 IFRB4 规则所获得的工件排序的前 F 个工件分别依次插入到 F 个工厂中; ②将第 $F+1$ 个工件以及之后的工件分别插入到 F 个工厂中所有可能的位置, 最终选取完工时间最小的位置; ③重复执行步骤②, 直到分配完所有工件为止。

随机生成方法。对于考虑序列相关准备时间的分布式混合零空闲置换流水车间调度问题中, 如果某个工厂在 $n > F$ 的情况下没有分配到工件, 显然是没有意义的。接下来将详细阐述随机初始化剩余种群个体的方法, 该方法保证每个工厂至少分配一个工件, 对于一个包含 n 个工件和 F 个工厂的 DMNIPFSP/SDST, 其具体描述如下: 首先, 随机生成一组含有 n 个工件的排列, 其次, 将所获得的工件排列进行分配给各个工厂, 其具体步骤如下: ①将通过随机生成方法所获得的工件排序的前 F 个工件分别依次插入到 F 个工厂中; ②将第 $F+1$ 个工件以及之后的工件插入到具有完工时间最小的工厂; ③重复执行步骤②, 直到分配完所有工件为止。

为了保证初始种群的质量和多样性，文献[24]、文献[25]以及文献[26]均表明，采用多种不同的启发式算法对种群个体进行初始化，有利于获得高质量的初始解，其余个体采用随机生成方法，有利于保持种群多样性。因此，本文采用四种初始化策略，分别为改进的 LR1 规则、LR2 规则、改进的 IFRB4 规则以及随机生成的初始化方法，其中，种群中前三个体分别采用 LR1 规则、LR2 规则、IFRB4 规则进行初始化，剩余种群个体采用随机生成方法。

3.3 种群更新机制

基本灰狼优化算法适用于解决连续优化问题，而 DMNIPFSP/SDST 是一种典型的离散优化问题。因此，在文献[27]所提出的离散化 GWO 迭代公式基础上，本文提出了一种多目标灰狼优化算法（MOGWO）求解 DMNIPFSP/SDST。具体更新步骤如下：

①随机生成 $[0,1]$ 之间的数 r ，如果 $r \leq 1/3$ ，那么将当前狼与 α 狼在同一位置上具有相同工件编码的工件删除，如果 $1/3 < r \leq 2/3$ ，那么将当前狼与 β 狼在同一位置上具有相同工件编码的工件删除，否则，将当前狼与 δ 狼在同一位置上具有相同工件编码的工件删除，得到参照狼工件序列 ω_1 ；

②通过收敛因子（A）与摆动因子（C），产生参照值 $K = |(A \cdot C)/2|$ ，其中，A 和 C 的取值范围均为 $[0,2]$ 之间的随机数；

③循环步骤①所获得的工件序列，每个工件生成一个对应的 $[0,1]$ 之间随机数，依次判断该随机数是否小于 K 值，如果小于 K 值，保留该工件位置，否则，将其删除，得到优化序列 ω_2 ；

④将步骤③所获得的优化序列 ω_2 处理步骤①所获得的参照狼工件序列 ω_1 ，其中工件序列中的每个工件都有 50% 概率执行交换（假设优化序列 ω_2 中某工件编号为 a ，所在工件对应的位置为 j ，寻找工件序列 ω_1 位置为 j 所对应的工件编号 b ，将工件序列 ω_1 中工件 a 与工件 b 进行交换），50% 概率执行插入（假设优化序列 ω_2 中某工件编号为 a ，所在工件对应的位置为 j ，寻找工件序列 ω_1 位置为 j 所对应的工件编号 b ，将工件序列 ω_1 中工件 b 插入到工件 a 之前），最终得到下一代个体。

F ₁	6	8	2
F ₂	3	5	
F ₃	4	1	7

F ₁	4	5		
F ₂	3	7		
F ₃	8	1	6	2

F ₁	3	1	6
F ₂	2	5	
F ₃	4	7	8

具体操作如图 2-图 5 所示。

(a) α 灰狼工件序列

(b) β 灰狼工件序列

(c) δ 灰狼工件序列

(d) ω 灰狼工件序列

图 2 不同灰狼的工件序列

F ₁	4	7	
F ₂	2	3	5
F ₃	6	1	8

执行步骤①，假设 $r \leq 1/3$ ，那么将当前狼与 α 狼在同一位置上具有相同工件编码的工件删除，具体详见图 3 所示。

F_1	4	7	
F_2			
F_3	6		8

图 3 步骤①结果

执行步骤②，假设 $K=1.3$ ；

执行步骤③，具体详见图 4 所示。

F_1	4(1.54)	7(0.76)	
F_2			
F_3	6(1.28)		8(0.92)

F_1		7	
F_2			
F_3	6		8

图 4 步骤③结果

执行步骤④，如果符合要求的工件都是满足交换概率，那么具体详见图 5 (a) 所示；如果工件

F_1	4	8	2
F_2	3	5	
F_3	6	1	7

F_1	7	8	2
F_2	3	5	
F_3	6	1	4

序号 7 和 6 是满足交换概率，而工件序号 8 满足插入概率，那么具体详见图 5 (b) 所示。

(a) 交换示意图

(b) 插入示意图

图 5 交换与插入示意图

3.4 邻域搜索

邻域搜索在多目标灰狼优化算法中起着非常重要的作用，有效的邻域结构可以防止算法过早地陷入局部最优以及提高算法的求解精度。已有相关研究表明，邻域搜索有助于获得求解调度问题的较优解^[24,28]。WANG 等^[24]开发了一种改进的布谷鸟搜索算法，引入了工件交换与插入的邻域结构，提高算法的搜索效率。ZHAO 等^[28]提出了一种离散差分进化算法，采用工厂内部与外部间的工件进行交换方式构建邻域解，有利于提高解的质量。然而，基于现有文献来看，局部搜索策略主要集中于工件的插入与交换，仍有待进一步研究。受 DENG 等^[29]和 ALLALI 等^[30]的启发，结合 DMNIPFSP/SDST 问题的特征，按照完工时间、能耗、拖期三个优化目标定义了三个种群，并且定义了三个关键工厂，即完工时间最大的工厂 (F_c)、能耗最大的工厂 (F_e) 以及拖期最大的工厂 (F_t)。

其中，对于完工时间的优化，本文提出了四种邻域搜索策略以改进在工厂内或工厂间使用交换或插入的解决方案。

LC1: 随机选择关键工厂 F_c 中一个工件，并将该工件与同工厂中的部分工件（该工厂中所有工件的一半）进行交换并保存可行解；

LC2: 随机选择关键工厂 F_c 中一个位置 p_1 ，然后从关键工厂以外的其他工厂中随机选择一个位置 p_2 ，将这两个位置对应的工件进行交换并保存可行解；

LC3: 随机选择关键工厂 F_c 中一个位置 p_1 ，然后从关键工厂以外的其他工厂中随机选择一个位置 p_2 ，将 p_2 位置所对应的工件插入到 p_1+1 以及将 p_1 位置所对应的工件插入到 p_2+1 并保存可行解；

LC4: 随机选择关键工厂 F_c 中一个位置 p_1 ，然后从关键工厂以外的其他工厂中随机选择一个位置 p_2 ，将 p_2 位置所对应的工件插入到 p_1-1 以及将 p_1 位置所对应的工件插入到 p_2-1 并保存可行解。

同理，对于能耗的优化，本文提出了四种邻域搜索策略以改进在工厂内或工厂间使用交换或插入的解决方案。

LE1: 随机选择关键工厂 F_c 中一个工件，并将该工件与同工厂中的部分工件（该工厂中所有工件的一半）进行交换并保存可行解；

LE2: 随机选择关键工厂 F_c 中一个位置 p_1 ，然后从关键工厂以外的其他工厂中随机选择一个位置 p_2 ，将这两个位置对应的工件进行交换并保存可行解；

LE3: 随机选择关键工厂 F_c 中一个位置 p_1 ，然后从关键工厂以外的其他工厂中随机选择一个位置 p_2 ，将 p_2 位置所对应的工件插入到 p_1+1 以及将 p_1 位置所对应的工件插入到 p_2+1 并保存可行解；

LE4: 随机选择关键工厂 F_c 中一个位置 p_1 ，然后从关键工厂以外的其他工厂中随机选择一个位置 p_2 ，将 p_2 位置所对应的工件插入到 p_1-1 以及将 p_1 位置所对应的工件插入到 p_2-1 并保存可行解。

对于拖期的优化，本文也提出了四种邻域搜索策略以改进在工厂内或工厂间使用交换或插入的解决方案。

LT1: 选择关键工厂 F_i 中最大拖期的工件 J_c , 然后从关键工厂 F_i 中随机选择 l 个工件 (不包括工件 J_c , l =该工厂中所有工件的一半), 将工件 J_c 分别与 l 个工件进行交换并保存可行解;

LT2: 选择关键工厂 F_i 中最大拖期的工件 J_c , 然后从关键工厂以外的其他每一个工厂中都随机选择一个工件 J_k , 将工件 J_c 与 J_k 分别进行交换并保存可行解;

LT3: 选择关键工厂 F_i 中最大拖期的工件 J_c , 然后将工件 J_c 分别插入到除关键工厂以外的其他工厂中随机选择的一个位置并保存可行解;

LT4: 选择关键工厂 F_i 中最大拖期的工件 J_c , 然后从关键工厂 F_i 中随机选择 l 个工件 (不包括工件 J_c , l =该工厂中所有工件的一半), 将工件 J_c 分别依次插入到 l 个工件前并保存可行解。

为了找到更好的解, 在上述完工时间、能耗以及拖期三个优化目标分别所对应的四种搜索策略的基础上, 提出了一种新的局部搜索, 以完工时间优化目标为例, 具体操作步骤如下:

①首先, 生成一个随机整数 $r \in [1, 4]$;

②假设 $r=2$, 那么执行邻域搜索的第二种策略, 若获得的新解优于当前解, 那么继续通过步骤①生成随机整数, 判定采用哪一种策略; 否则, 从剩余的三种策略中, 随机选择一种策略执行, 直到以上四种策略所得出的解均劣于当前解, 退出邻域搜索。

3.5 种群协同

由于每个种群在位置更新都集中于一个目标, 极易造成三个目标之间的不平衡。为了很好的平衡最大完工时间、总能耗和总拖期三个目标, 本文设计了三个种群 (PC、PE 和 PT), 其中, 种群 PC 主要以最小化最大完工时间为目标进行位置更新; 种群 PE 主要以最小化总能耗为目标进行位置更新; 种群 PT 主要以最小化总拖期为目标进行位置更新。当每隔 L 次迭代时, 三个种群进行相互交换个体, 具体操作流程如下: 对于 PC 种群而言, 每个个体按照最大完工时间进行升序排列, 取出前三分之一的个体分别转移到 PE 和 PT 种群; 同理, 对于 PE 种群而言, 每个个体按照总能耗进行升序排列, 取出前三分之一的个体分别转移到 PC 和 PT 种群; 对于 PT 种群而言, 每个个体按照总拖期进行升序排列, 取出前三分之一的个体分别转移到 PC 和 PE 种群。经过上述交换后, 每个种群将会多出三分之一的个体, 采用 3.6 节精英策略保留较优的个体。

3.6 精英策略

本文以 NSGA-II 算法中的非支配排序和拥挤度距离方法^[31]为标准设计了一种新的精英策略，其评判依据：非支配排序值小的解决方案比非支配排序值大的解决方案好，为了保存种群的多样性，拥挤度距离较大的解优于拥挤度距离较小的解，因此，一个高质量的 Pareto 最优解集通常是一组非支配排序值小且拥挤度距离较大的解。在 MOGWO 的种群更新机制过程：首先，将种群个体和新产生的个体合并，对所有个体进行非支配排序和计算相关的拥挤度距离；其次，优先选择非支配排序值小的个体，当非支配排序值相同时，选择拥挤度距离值大的个体；最后，根据上述的精英策略选择下一代种群。

3.7 MOGWO 算法框架

针对以最小化最大完工时间、总能耗和总拖期为优化目标的序列相关准备时间的分布式混合零空闲置换流水车间调度问题，提出了一种融合不同初始化策略、种群更新机制、邻域搜索、种群协同以及精英策略的多目标灰狼优化算法，改进算法的伪代码具体如图 6 所示。首先，提出了三种初始化策略和随机生成相结合的方法初始化种群，有利于提高种群的质量和多样性；其次，在位置更新阶段，针对基本灰狼优化算法无法直接应用于求解离散车间调度问题的不足，提出了一种离散化种群更新机制，使算法能够直接在离散调度空间中进行搜索，并且很好地平衡了算法的全局探索和局部开发能力；为了进一步提高解的质量和求解精度，针对不同的优化目标，各提出了四种邻域策略进行局部搜索；由于每个种群在位置更新都集中于一个目标，极易造成三个目标之间的不平衡，设计了种群协同机制平衡三个优化目标，通过精英策略选择下一代种群；最后，直到满足终止条件，输出最优的解决方案。

MOGWO 算法:

输入: 种群大小 (N)；种群交互间隔 (L)；终止条件

输出: 最优解决方案

开始

1: 针对不同的优化目标，划分为三个子种群

2: 采用三种初始化策略和随机生成的方法初始化种群

3: **while** 不满足终止条件 **do**

4: 根据第 3.3 节的种群更新机制更新三个子种群

5: 根据第 3.4 节的邻域搜索选择相应的邻域策略进行局部搜索

6: 根据第 3.5 节的种群协同机制进行种群间相互交换，采用第 3.6 节精英策略选择下一代种群

7: **end while**

8: 合并三个子种群，采用非支配排序输出最优解决方案

结束

图 6 MOGWO 算法伪代码

4 实验结果分析

4.1 测试实例

为了评估所提出的 MOGWO 算法性能, 本文参考了 RUIZ 等^[32]、RUIZ 和 STUTZLE^[33]文献中数据集, 问题实例包含四种不同的序列相关准备时间 (SDST-10、SDST-50、SDST-100、SDST-125)。其中, 工件数量和机器之间的组合: $\{n,m\}=\{20,5\},\{20,10\},\{20,20\},\{50,5\},\{50,10\},\{50,20\},\{100,5\},\{100,10\},\{100,20\},\{200,10\},\{200,20\}$, 选取每种规模中的最后一个算例进行实验, $F=\{2,4,6\}$, 不同机器上的加工速度 $v_i=\{1.0,2.0,3.0\}$, 单位时间的加工能耗 $PP_i=4\times v_i^2 kw$, 单位时间的序列相关准备时间能耗 $SP_i=[1kw,2kw]$, 单位时间的空闲能耗 $IP_i=0.25\times PP_i kw$ 。另外, 由于本文解决的是混合零空闲置换流水车间调度问题, 所以考虑了三种不同的混合零空闲约束场景, 分别为: ①随机选择 25% 的机器为零空闲; ②随机选择 50% 的机器为零空闲; ③随机选择 75% 的机器为零空闲, 综上, 基准测试中的算例总数为 $4\times 3\times 3\times 11=396$ 。

与文献[34]相同, 采用反世代距离 (Inverted generational distance, IGD)、覆盖率 (Coverage metric, C-metric) 评估本文所提出的多目标灰狼优化算法 (MOGWO) 求解 DMNIPFSP/SDST 在每一个算例的性能, IGD 评价指标具体的计算公式如 (20) 所示:

$$IGD(s, S^*) = \frac{1}{|S^*|} \sum_{s \in S^*} distance(s, S^*) \quad (20)$$

其中, $distance(s, S^*)$ 表示最优解集 S^* 中的点 s 到某一特定算法获得的非支配解集 S^* 的最近距离, 其中 $|S^*|$ 表示 S^* 中解的数目, 最优解集 S^* 是合并所有算法获得的非支配解集并剔除支配解得到的。

C-metric 指标被用来度量两个算法的非支配解集之间的支配关系, 如 $C(A, B)$ 表示 B 算法所求解的非支配解中有多少比例的解被 A 算法的非支配解集所支配, 具体的计算公式如 (21) 所示:

$$C(A, B) = \frac{|\{b \in B : \exists a \in A, a \succ b\}|}{|B|} \quad (21)$$

基于此, 将多目标灰狼优化算法与改进的非支配排序遗传算法 (Improved non-dominated sorting genetic algorithm II, INSGA-II)^[35]、多目标混合迭代贪婪算法 (Multi-objective hybrid iterated greedy, MOHIG)^[36]、竞争模因算法 (Competitive memetic algorithm, CMA)^[29] 进行对比实验, 所有对比算法的终止时间设置为 $n \times m \times f \times C$ 毫秒, C 的取值为 5。工件的交货期设置为 $d_j = P_j \times (1 + rand(0,1))$, P_j 表示为工件 j 在所有机器上加工时间总和。所有算法在测试算例独立运行 10 次。

4.2 参数设置

由于控制参数的取值对算法的性能有着重要的影响,若能选取最佳控制参数值有助于提高算法的性能。因此,本文所提出的多目标灰狼优化算法的控制参数分别为种群大小(N)和种群交互间隔(L)。为了确定这2个参数的最佳组合,设计了 Taguchi 方法^[37],每个参数设计了4种不同的水平因子,各影响因素参数水平详见表2所示,采用 $L_{16}(4^2)$ 正交表在混合零空闲约束条件类型为第①种的 $20 \times 20 \times 2$ 的 SDST-100 算例进行实验,每种参数组合下的测试算例均独立运行10次,每个参数组合下的运行时间为 $20 \times 20 \times 2 \times 5$ 毫秒,选择反世代距离(IGD)指标进行评价,具体参数组合详见表3所示和图7所示。

表2 参数水平表

水平	N	L
1	20	10
2	30	15
3	40	20
4	50	25

表3 正交表和实验结果

组合	N	L	IGD
1	20(1)	10(1)	5043.33
2	20(1)	15(2)	6684.34
3	20(1)	20(3)	4255.75
4	20(1)	25(4)	4101.73
5	30(2)	10(1)	5651.10
6	30(2)	15(2)	2056.07
7	30(2)	20(3)	5549.18
8	30(2)	25(4)	5164.86
9	40(3)	10(1)	4143.86
10	40(3)	15(2)	3097.66
11	40(3)	20(3)	2623.52
12	40(3)	25(4)	4031.79

13	50(4)	10(1)	3394.69
14	50(4)	15(2)	7694.51
15	50(4)	20(3)	4221.34
16	50(4)	25(4)	5229.61

表 4 响应表

水平	N	L
1	5021.2880	4558.2457
2	4605.3024	4883.1463
3	3474.2068	4162.4464
4	5135.0401	4631.9988
极差	1660.8334	720.6999
等级	1	2

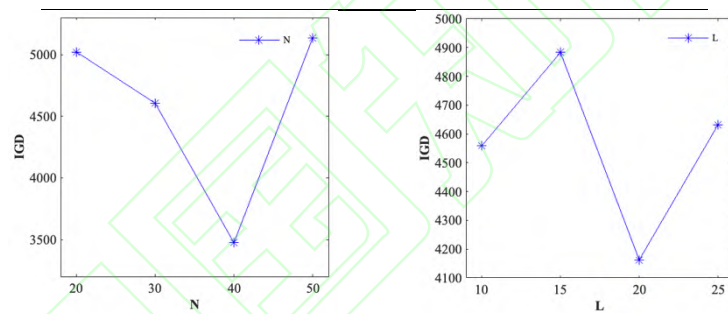


图 7 各参数水平响应趋势

表 4 和图 7 展示了各因素水平的响应值，最优值加粗显示，种群大小 (N) 极差最大，说明种群大小变化对所求解问题的结果影响最大，由于种群大小 (N) 决定了搜索空间的范围，若该值设置过小，容易造成搜索范围过于局限，不能很好探索整个空间；若该值设置过大，那么会耗费过多时间进行求解，降低算法的计算效率和收敛性。种群交互间隔 (L) 极差最小，说明 L 参数变化对所求解问题的结果影响最小，但合理设置 L 值有助于提高算法对问题的求解效果。综上，表 4 中当 N 和 L 分别取第三水平和第三水平时，所求解问题的结果达到最优，因此，这两个参数的最佳组合为 $N=40$ ， $L=20$ 。

4.3 邻域搜索策略有效性

为了验证所提出的邻域搜索策略的有效性，将带有邻域搜索策略的多目标灰狼优化算法（命名为 LMOGWO）与没有邻域搜索策略的多目标灰狼优化算法（命名为 NMOGWO）进行比较，两种

算法的其他组成部分相同。通过使用反世代距离（IGD）和覆盖率（C-metric）两个评价指标对这两种算法在不同算例的结果进行对比分析。最优结果加粗显示，具体详见表 5-12 所示。

表 5 在 SDST-10 约束下 LMOGWO 和 NMOGWO 的 IGD 值

类型	①		②		③	
	LMOGWO	NMOGWO	LMOGWO	NMOGWO	LMOGWO	NMOGWO
$F=2$	2.40E+04	6.94E+05	7.71E+04	6.53E+05	4.11E+04	7.17E+05
$F=4$	7.82E+04	6.59E+05	7.18E+04	6.78E+05	1.90E+05	5.89E+05
$F=6$	3.61E+04	7.30E+05	7.24E+04	7.02E+05	1.73E+05	6.29E+05
均值	4.61E+04	6.94E+05	7.38E+04	6.78E+05	1.35E+05	6.45E+05
$n=20$	1.93E+04	1.42E+05	3.28E+04	1.42E+05	8.00E+03	1.65E+05
$n=50$	2.83E+04	2.51E+05	2.14E+04	2.53E+05	2.72E+04	2.83E+05
$n=100$	4.54E+04	7.96E+05	8.05E+04	8.56E+05	7.69E+04	8.60E+05
$n=200$	1.14E+05	2.04E+06	2.04E+05	1.85E+06	5.73E+05	1.59E+06
均值	5.18E+04	8.07E+05	8.47E+04	7.75E+05	1.71E+05	7.25E+05
$m=5$	2.93E+04	1.25E+05	1.80E+04	1.35E+05	3.47E+04	1.18E+05
$m=10$	4.06E+04	5.628E+05	1.55E+05	4.16E+05	1.44E+05	4.52E+05
$m=20$	6.43E+04	1.25E+06	3.46E+04	1.35E+06	2.00E+05	1.23E+06
均值	4.47E+04	6.46E+05	6.92E+04	6.34E+05	1.26E+05	6.00E+05

表 6 在 SDST-50 约束下 LMOGWO 和 NMOGWO 的 IGD 值

类型	①		②		③	
	LMOGWO	NMOGWO	LMOGWO	NMOGWO	LMOGWO	NMOGWO
$F=2$	1.47E+04	8.32E+05	1.93E+04	9.06E+05	9.98E+04	7.88E+05
$F=4$	5.68E+04	8.13E+05	2.10E+04	9.74E+05	1.33E+05	7.53E+05
$F=6$	4.35E+04	8.21E+05	1.30E+05	8.31E+05	1.18E+05	8.32E+05
均值	3.83E+04	8.22E+05	5.68E+04	9.04E+05	1.17E+05	7.91E+05
$n=20$	3.48E+03	2.16E+05	3.85E+04	2.05E+05	5.70E+04	1.80E+05
$n=50$	1.11E+02	3.53E+05	3.45E+02	3.78E+05	6.47E+04	3.27E+05
$n=100$	1.37E+05	7.30E+05	5.35E+04	8.42E+05	7.39E+04	7.52E+05
$n=200$	0.00E+00	2.57E+06	1.74E+05	2.83E+06	3.49E+05	2.46E+06
均值	3.51E+04	9.67E+05	6.66E+04	1.06E+06	1.36E+05	9.30E+05
$m=5$	7.89E+03	1.16E+05	1.70E+04	9.59E+04	1.51E+04	1.03E+05
$m=10$	1.28E+04	5.82E+05	3.22E+04	6.01E+05	6.73E+04	5.14E+05
$m=20$	8.67E+04	1.59E+06	1.11E+05	1.81E+06	2.43E+05	1.58E+06
均值	3.58E+04	7.63E+05	5.34E+04	8.36E+05	1.08E+05	7.32E+05

表 7 在 SDST-100 约束下 LMOGWO 和 NMOGWO 的 IGD 值

类型	①		②		③	
	LMOGWO	NMOGWO	LMOGWO	NMOGWO	LMOGWO	NMOGWO

$F=2$	0.00E+00	9.08E+05	8.27E+04	7.37E+05	2.11E+04	8.79E+05
$F=4$	2.05E+04	8.70E+05	3.86E+04	8.53E+05	9.59E+04	7.38E+05
$F=6$	1.41E+05	7.26E+05	8.25E+03	8.37E+05	7.49E+04	8.09E+05
均值	5.38E+04	8.35E+05	4.32E+04	8.09E+05	6.40E+04	8.09E+05
$n=20$	0.00E+00	2.20E+05	7.83E+03	2.29E+05	4.73E+03	2.51E+05
$n=50$	0.00E+00	4.03E+05	2.36E+04	3.82E+05	5.59E+03	3.62E+05
$n=100$	9.65E+04	8.48E+05	1.15E+05	7.14E+05	1.46E+05	6.78E+05
$n=200$	1.51E+05	2.39E+06	1.79E+04	2.46E+06	1.18E+05	2.51E+06
均值	6.19E+04	9.65E+05	4.11E+04	9.46E+05	6.86E+04	9.50E+05
$m=5$	4.77E+03	1.33E+05	8.29E+03	1.33E+05	1.99E+04	1.09E+05
$m=10$	1.52E+04	4.40E+05	1.50E+04	4.38E+05	5.91E+04	3.92E+05
$m=20$	1.29E+05	1.76E+06	9.75E+04	1.69E+06	1.02E+05	1.75E+06
均值	4.97E+04	7.78E+05	4.03E+04	7.54E+05	6.03E+04	7.50E+05

表 8 在 SDST-125 约束下 LMOGWO 和 NMOGWO 的 IGD 值

类型	①		②		③	
	LMOGWO	NMOGWO	LMOGWO	NMOGWO	LMOGWO	NMOGWO
$F=2$	6.51E+01	9.17E+05	8.43E+04	9.83E+05	3.60E+04	8.09E+05
$F=4$	4.52E+04	9.13E+05	5.77E+04	8.45E+05	6.00E+04	7.99E+05
$F=6$	5.23E+04	9.03E+05	6.45E+04	9.58E+05	8.17E+04	7.77E+05
均值	3.25E+04	9.11E+05	6.88E+04	9.29E+05	5.92E+04	7.95E+05
$n=20$	0.00E+00	2.32E+05	4.48E+03	2.34E+05	0.00E+00	2.08E+05
$n=50$	7.96E+01	2.79E+05	4.24E+04	3.25E+05	3.29E+03	3.48E+05
$n=100$	6.39E+04	8.89E+05	1.81E+05	6.60E+05	7.70E+04	6.68E+05
$n=200$	8.28E+04	2.91E+06	3.75E+04	3.28E+06	2.06E+05	2.54E+06
均值	3.67E+04	1.08E+06	6.63E+04	1.12E+06	7.16E+04	9.41E+05
$m=5$	9.13E+03	1.20E+05	2.25E+04	1.06E+05	2.50E+04	9.94E+04
$m=10$	4.15E+04	4.91E+05	4.17E+04	4.69E+05	1.12E+05	3.96E+05
$m=20$	4.11E+04	1.92E+06	1.31E+05	2.00E+06	3.21E+04	1.72E+06
均值	3.06E+04	8.44E+05	6.51E+04	8.58E+05	5.64E+04	7.38E+05

表 9 在 SDST-10 约束下 LMOGWO 和 NMOGWO 的 C-metric 值

类型	①		②		③	
	C (LM, NM)	C (NM, LM)	C (LM, NM)	C (NM, LM)	C (LM, NM)	C (NM, LM)
$F=2$	0.85	0.00	0.65	0.00	0.95	0.00
$F=4$	0.67	0.00	0.84	0.00	0.60	0.00

$F=6$	0.86	0.00	0.59	0.00	0.66	0.00
均值	0.79	0.00	0.69	0.00	0.74	0.00
$n=20$	0.81	0.00	0.53	0.00	0.91	0.00
$n=50$	0.80	0.00	0.89	0.00	0.72	0.00
$n=100$	0.85	0.00	0.74	0.00	0.73	0.00
$n=200$	0.67	0.00	0.58	0.00	0.51	0.00
均值	0.78	0.00	0.69	0.00	0.72	0.00
$m=5$	0.72	0.00	0.70	0.00	0.72	0.00
$m=10$	0.82	0.00	0.57	0.00	0.69	0.00
$m=20$	0.82	0.00	0.81	0.00	0.81	0.00
均值	0.79	0.00	0.69	0.00	0.74	0.00

注：LM 表示 LMOGWO；NM 表示 NMOGWO。下同。

表 10 在 SDST-50 约束下 LMOGWO 和 NMOGWO 的 C-metric 值

类型	①		②		③	
	C (LM, NM)	C (NM, LM)	C (LM, NM)	C (NM, LM)	C (LM, NM)	C (NM, LM)
$F=2$	0.86	0.00	0.83	0.00	0.55	0.00
$F=4$	0.74	0.00	0.61	0.00	0.53	0.00
$F=6$	0.89	0.00	0.70	0.00	0.58	0.00
均值	0.83	0.00	0.71	0.00	0.55	0.00
$n=20$	0.89	0.00	0.73	0.00	0.50	0.00
$n=50$	0.93	0.00	0.89	0.00	0.41	0.00
$n=100$	0.56	0.00	0.44	0.00	0.72	0.00
$n=200$	1.00	0.00	0.83	0.00	0.58	0.00
均值	0.85	0.00	0.72	0.00	0.55	0.00
$m=5$	0.78	0.00	0.58	0.00	0.78	0.00
$m=10$	0.86	0.00	0.67	0.00	0.54	0.00
$m=20$	0.83	0.00	0.86	0.00	0.39	0.00
均值	0.82	0.00	0.70	0.00	0.57	0.00

表 11 在 SDST-100 约束下 LMOGWO 和 NMOGWO 的 C-metric 值

类型	①		②		③	
	C (LM, NM)	C (NM, LM)	C (LM, NM)	C (NM, LM)	C (LM, NM)	C (NM, LM)
$F=2$	1.00	0.00	0.50	0.00	0.73	0.00
$F=4$	0.86	0.00	0.91	0.00	0.65	0.00
$F=6$	0.82	0.00	0.82	0.00	0.61	0.00
均值	0.89	0.00	0.74	0.00	0.66	0.00
$n=20$	1.00	0.00	0.78	0.00	0.85	0.00
$n=50$	1.00	0.00	0.72	0.00	0.85	0.00
$n=100$	0.72	0.00	0.67	0.00	0.28	0.00

$n=200$	0.83	0.00	0.83	0.00	0.67	0.00
均值	0.89	0.00	0.75	0.00	0.66	0.00
$m=5$	0.89	0.00	0.83	0.00	0.70	0.00
$m=10$	0.96	0.00	0.67	0.00	0.54	0.00
$m=20$	0.83	0.00	0.75	0.00	0.75	0.00
均值	0.89	0.00	0.75	0.00	0.66	0.00

表 12 在 SDST-125 约束下 LMOGWO 和 NMOGWO 的 C-metric 值

类型	①		②		③	
	C (LM, NM)	C (NM, LM)	C (LM, NM)	C (NM, LM)	C (LM, NM)	C (NM, LM)
$F=2$	0.95	0.00	0.68	0.00	0.71	0.00
$F=4$	0.91	0.00	0.68	0.00	0.73	0.00
$F=6$	0.82	0.00	0.80	0.00	0.77	0.00
均值	0.89	0.00	0.72	0.00	0.74	0.00
$n=20$	1.00	0.00	0.97	0.00	1.00	0.00
$n=50$	0.94	0.00	0.78	0.00	0.81	0.00
$n=100$	0.78	0.00	0.33	0.00	0.50	0.00
$n=200$	0.83	0.00	0.83	0.00	0.58	0.00
均值	0.89	0.00	0.73	0.00	0.72	0.00
$m=5$	0.89	0.00	0.56	0.00	0.61	0.00
$m=10$	0.88	0.00	0.73	0.00	0.65	0.00
$m=20$	0.92	0.00	0.83	0.00	0.92	0.00
均值	0.90	0.00	0.71	0.00	0.73	0.00

表 5-12 的结果可知，在不同序列相关准备时间以及不同零空闲约束场景下的所有算例，LMOGWO 所求解出的 IGD 值均比 NMOGWO 所求解出的 IGD 值小，说明包含邻域搜索策略的多目标灰狼优化算法（LMOGWO）求得解的收敛性和分布性更好；另外，对于不同零空闲约束场景下的所有算例的 C（LM，NM）值均比 C（NM，LM）值大，表明 NMOGWO 的大部分解被 LMOGWO 的非支配解支配。为了进一步说明包含邻域搜索策略的多目标灰狼优化算法（LMOGWO）在性能上的优越性，通过对 LMOGWO 和 NMOGWO 在不同算例下的测试结果进行方差分析以更为直观的形式呈现出这两种算法之间所存在的性能差异。图 8 显示出两种算法的均值变化线以及 95%置信度下的 Turkey's HSD 检验的置信区间，从图 8 可看出，不同邻域搜索策略的引入对多目标灰狼优化算法的性能具有显著的积极影响，从而也进一步证明了邻域搜索策略有助于改善多目标灰狼优化算法的性能，并且也可得出基于邻域搜索策略的多目标灰狼优化算法可以有效地解决这一问题。

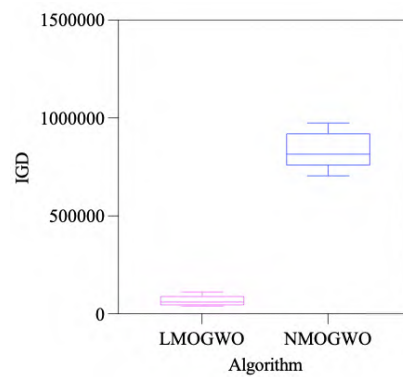


图 8 LMOGWO 与 NMOGWO 的方差分析图

4.4 与相关算法的比较

为了验证 MOGWO 与其他算法的性能,表 13-16 分别展示了 SDST-10、SDST-50、SDST-100、SDST-125 时,按工厂、工件和机器分组的多目标灰狼优化算法(MOGWO)与三种对比算法在所有算例的 IGD 计算结果。从表 13-16 可以发现,无论在何种序列相关准备时间下的不同零空闲约束场景,本文所提出的 MOGWO 在求解绝大部分算例上所获的结果均优于其他对比算法,尤其是随着工件数量的增加,MOGWO 表现出更加明显的优势。具体而言,按工件分组来看,当序列相关准备时间为 SDST-10 时,MOGWO 在三种不同零空闲中所有算例的 IGD 均值分别为 $2.03\text{E}+03$ 、 $3.81\text{E}+03$ 以及 $5.19\text{E}+03$,远小于 INSGA-II 以及 CMA;当序列相关准备时间为 SDST-50 时,MOGWO 在三种不同零空闲中所有算例的 IGD 均值分别为 $1.25\text{E}+04$ 、 $7.92\text{E}+03$ 以及 $5.16\text{E}+03$,远小于 INSGA-II、MOHIG 以及 CMA;当序列相关准备时间为 SDST-100 时,MOGWO 在三种不同零空闲中所有算例的 IGD 均值分别为 $9.12\text{E}+03$ 、 $6.92\text{E}+03$ 以及 $6.62\text{E}+03$,远小于 INSGA-II、MOHIG 以及 CMA;当序列相关准备时间为 SDST-125 时,MOGWO 在三种不同零空闲中所有算例的 IGD 均值分别为 $9.24\text{E}+03$ 、 $1.17\text{E}+04$ 以及 $7.49\text{E}+03$,远小于 INSGA-II、MOHIG 以及 CMA,表明本文所提出的 MOGWO 具有一定的优越性,这可能的原因是改进的初始化策略以及随机生成相结合的初始化方法,使初始解均匀地分布在不同的区域,有利于提高解的质量和多样性;其次,采用了离散化的种群更新机制,使算法能够直接在离散调度空间中进行搜索,并且设计了种群协同机制平衡三个优化目标,从而极大地增强了全局探索和局部开发能力;此外,针对不同的优化目标,提出了不同的邻域搜索策略,避免了搜索策略单一导致算法易陷入局部最优问题,并且进一步提高了解的质量和求解精度。

表 13 在 SDST-10 约束下不同算法的 IGD 值

类型	①				②				③			
	INSGA-II	MOHIG	CMA	MOGWO	INSGA-II	MOHIG	CMA	MOGWO	INSGA-II	MOHIG	CMA	MOGWO
F=2	9.72E+03	5.23E+03	3.12E+04	2.26E+03	1.27E+04	9.21E+03	2.58E+04	3.58E+03	6.82E+03	7.63E+03	2.21E+04	2.49E+03

$F=4$	9.73E+03	3.48E+03	2.81E+04	1.34E+03	7.93E+03	6.60E+03	1.98E+04	2.94E+03	5.95E+03	5.90E+03	1.99E+04	3.20E+03
$F=6$	8.31E+03	2.50E+03	2.71E+04	1.63E+03	8.68E+03	5.50E+03	2.34E+04	2.45E+03	8.44E+03	6.38E+03	2.26E+04	2.21E+03
均值	9.25E+03	3.74E+03	2.88E+04	1.74E+03	9.77E+03	7.10E+03	2.30E+04	2.99E+03	7.07E+03	6.64E+03	2.15E+04	2.63E+03
$n=20$	1.16E+03	1.40E+03	1.02E+03	4.59E+02	1.29E+03	2.29E+03	2.15E+03	4.19E+02	1.22E+03	1.35E+03	1.24E+03	4.66E+02
$n=50$	3.20E+03	1.89E+03	6.43E+03	7.93E+02	3.13E+03	2.72E+03	5.05E+03	5.76E+02	2.85E+03	2.60E+03	4.13E+03	8.68E+02
$n=100$	1.14E+04	2.76E+03	3.36E+04	1.69E+03	1.43E+04	2.44E+03	3.36E+04	1.46E+03	1.75E+04	1.86E+04	6.10E+04	9.50E+03
$n=200$	2.73E+04	1.15E+04	9.67E+04	5.16E+03	2.56E+04	2.79E+04	6.53E+04	1.28E+04	1.73E+04	1.80E+04	6.69E+04	9.91E+03
均值	1.08E+04	4.39E+03	3.44E+04	2.03E+03	1.11E+04	8.84E+03	2.65E+04	3.81E+03	9.72E+03	1.01E+04	3.33E+04	5.19E+03
$m=5$	1.36E+03	1.55E+03	3.72E+03	5.52E+02	1.12E+03	1.28E+03	3.41E+03	5.39E+02	2.15E+03	3.29E+03	2.69E+03	1.28E+03
$m=10$	9.07E+03	2.91E+03	2.03E+04	1.31E+03	1.02E+04	1.07E+04	1.40E+04	4.99E+03	5.43E+03	5.49E+03	1.46E+04	2.33E+03
$m=20$	1.54E+04	6.20E+03	5.60E+04	3.07E+03	1.58E+04	7.87E+03	4.67E+04	2.83E+03	1.24E+04	1.03E+04	4.25E+04	3.96E+03
均值	8.61E+03	3.55E+03	2.67E+04	1.64E+03	9.04E+03	6.62E+03	2.14E+04	2.79E+03	6.66E+03	6.36E+03	1.99E+04	2.52E+03

表 14 在 SDST-50 约束下不同算法的 IGD 值

类型	①				②				③			
	INSGA-II	MOHIG	CMA	MOGWO	INSGA-II	MOHIG	CMA	MOGWO	INSGA-II	MOHIG	CMA	MOGWO
$F=2$	3.87E+04	2.23E+04	1.14E+05	1.44E+04	1.67E+04	1.77E+04	1.14E+05	8.93E+03	2.74E+04	1.43E+04	9.96E+04	5.77E+03
$F=4$	2.48E+04	1.47E+04	6.93E+04	9.71E+03	2.19E+04	1.26E+04	8.54E+04	4.83E+03	1.28E+04	8.60E+03	6.59E+04	2.91E+03
$F=6$	1.74E+04	1.49E+04	6.26E+04	5.09E+03	1.33E+04	9.76E+03	6.99E+04	4.98E+03	1.45E+04	8.05E+03	6.37E+04	3.35E+03
均值	2.70E+04	1.73E+04	8.20E+04	9.73E+03	1.73E+04	1.34E+04	8.98E+04	6.25E+03	1.82E+04	1.03E+04	7.64E+04	4.01E+03
$n=20$	3.29E+03	2.45E+03	1.71E+03	8.82E+02	2.90E+03	1.96E+03	2.46E+03	6.57E+02	3.65E+03	3.34E+03	2.31E+03	8.71E+02
$n=50$	5.58E+03	5.39E+03	8.97E+03	1.57E+03	3.87E+03	5.14E+03	7.66E+03	1.80E+03	6.21E+03	5.65E+03	1.25E+04	6.90E+02
$n=100$	3.12E+04	8.39E+03	8.10E+04	4.88E+03	1.98E+04	9.60E+03	8.36E+04	2.94E+03	1.41E+04	1.06E+04	6.49E+04	1.37E+03
$n=200$	8.83E+04	7.09E+04	3.14E+05	4.26E+04	5.52E+04	4.84E+04	3.52E+05	2.63E+04	6.43E+04	2.74E+04	3.01E+05	1.77E+04
均值	3.21E+04	2.18E+04	1.01E+05	1.25E+04	2.04E+04	1.63E+04	1.11E+05	7.92E+03	2.21E+04	1.17E+04	9.52E+04	5.16E+03
$m=5$	7.65E+03	2.83E+03	8.81E+03	2.03E+03	2.60E+03	2.89E+03	4.20E+03	1.08E+03	2.21E+03	2.04E+03	3.39E+03	7.78E+02
$m=10$	2.67E+04	1.79E+04	6.48E+04	8.03E+03	1.32E+04	1.49E+04	6.47E+04	4.68E+03	2.13E+04	9.63E+03	5.25E+04	6.16E+03
$m=20$	4.18E+04	2.76E+04	1.54E+05	1.73E+04	3.23E+04	1.96E+04	1.79E+05	1.17E+04	2.71E+04	1.72E+04	1.55E+05	4.29E+03
均值	2.54E+04	1.61E+04	7.59E+04	9.12E+03	1.60E+04	1.25E+04	8.26E+04	5.82E+03	1.69E+04	9.62E+03	7.03E+04	3.74E+03

表 15 在 SDST-100 约束下不同算法的 IGD 值

类型	①				②				③			
	INSGA-II	MOHIG	CMA	MOGWO	INSGA-II	MOHIG	CMA	MOGWO	INSGA-II	MOHIG	CMA	MOGWO
$F=2$	2.98E+04	3.41E+04	2.03E+05	1.03E+04	3.56E+04	1.66E+04	1.74E+05	9.56E+03	4.16E+04	1.58E+04	1.91E+05	4.20E+03
$F=4$	3.37E+04	2.06E+04	1.36E+05	6.03E+03	3.25E+04	1.57E+04	1.28E+05	3.83E+03	1.70E+04	1.35E+04	9.80E+04	8.17E+03
$F=6$	2.24E+04	1.42E+04	9.83E+04	5.16E+03	2.47E+04	1.60E+04	9.88E+04	4.31E+03	2.82E+04	1.30E+04	1.02E+05	3.52E+03
均值	2.86E+04	2.30E+04	1.46E+05	7.16E+03	3.09E+04	1.61E+04	1.34E+05	5.90E+03	2.89E+04	1.41E+04	1.30E+05	5.30E+03
$n=20$	5.02E+03	5.05E+03	5.86E+03	9.59E+02	4.28E+03	3.22E+03	2.23E+03	8.41E+02	5.92E+03	5.22E+03	4.48E+03	1.15E+03
$n=50$	1.20E+04	1.11E+04	1.57E+04	1.85E+03	7.18E+03	6.63E+03	1.41E+04	2.05E+03	7.09E+03	7.44E+03	1.44E+04	1.17E+03
$n=100$	1.92E+04	6.80E+03	1.01E+05	2.98E+03	3.37E+04	2.06E+04	1.00E+05	6.58E+03	3.26E+04	1.65E+04	8.66E+04	3.06E+03

$n=200$	1.03E+05	9.20E+04	6.18E+05	3.07E+04	1.03E+05	4.29E+04	5.59E+05	1.82E+04	9.06E+04	3.39E+04	5.59E+05	2.11E+04
均值	3.48E+04	2.87E+04	1.85E+05	9.12E+03	3.70E+04	1.83E+04	1.69E+05	6.92E+03	3.41E+04	1.58E+04	1.66E+05	6.62E+03
$m=5$	5.08E+03	2.47E+03	1.05E+04	9.18E+02	5.02E+03	4.78E+03	9.10E+03	1.74E+03	3.57E+03	1.69E+03	8.04E+03	3.80E+02
$m=10$	2.36E+04	2.11E+04	9.12E+04	6.90E+03	2.09E+04	1.83E+04	9.82E+04	1.11E+04	2.92E+04	1.11E+04	1.00E+05	6.68E+03
$m=20$	5.14E+04	4.03E+04	3.02E+05	1.21E+04	6.04E+04	2.24E+04	2.62E+05	3.83E+03	4.77E+04	2.65E+04	2.52E+05	7.60E+03
均值	2.67E+04	2.13E+04	1.35E+05	6.64E+03	2.88E+04	1.52E+04	1.23E+05	5.56E+03	2.68E+04	1.31E+04	1.20E+05	4.89E+03

表 16 在 SDST-125 约束下不同算法的 IGD 值

类型	①				②				③			
	INSGA-II	MOHIG	CMA	MOGWO	INSGA-II	MOHIG	CMA	MOGWO	INSGA-II	MOHIG	CMA	MOGWO
$F=2$	5.05E+04	1.88E+04	2.49E+05	6.35E+03	9.26E+04	4.83E+04	3.76E+05	9.03E+03	3.26E+04	4.01E+04	2.42E+05	8.85E+03
$F=4$	2.63E+04	1.75E+04	1.53E+05	4.45E+03	2.02E+04	2.18E+04	1.98E+05	1.26E+04	3.24E+04	2.10E+04	1.53E+05	3.46E+03
$F=6$	1.95E+04	2.10E+04	1.18E+05	1.15E+04	1.98E+04	1.93E+04	1.42E+05	6.78E+03	2.73E+04	2.12E+04	1.47E+05	6.86E+03
均值	3.21E+04	1.91E+04	1.73E+05	7.43E+03	4.42E+04	2.98E+04	2.39E+05	9.47E+03	3.08E+04	2.74E+04	1.81E+05	6.39E+03
$n=20$	3.77E+03	5.10E+03	4.58E+03	1.58E+03	4.39E+03	8.45E+03	5.09E+03	1.32E+03	5.34E+03	6.48E+03	5.50E+03	1.76E+03
$n=50$	9.71E+03	1.08E+04	1.70E+04	1.91E+03	1.28E+04	1.18E+04	1.77E+04	2.24E+03	1.21E+04	1.68E+04	2.59E+04	1.20E+03
$n=100$	3.83E+04	2.45E+04	1.24E+05	4.27E+03	3.55E+04	1.72E+04	1.19E+05	6.92E+03	4.35E+04	2.91E+04	1.41E+05	7.38E+03
$n=200$	9.88E+04	4.45E+04	7.35E+05	2.92E+04	1.64E+05	1.08E+05	1.10E+06	3.63E+04	7.77E+04	7.26E+04	7.36E+05	1.96E+04
均值	3.76E+04	2.12E+04	2.20E+05	9.24E+03	5.42E+04	3.64E+04	3.10E+05	1.17E+04	3.47E+04	3.12E+04	2.27E+05	7.49E+03
$m=5$	5.90E+03	4.77E+03	7.40E+03	1.38E+03	6.57E+03	4.67E+03	7.63E+03	2.54E+03	6.70E+03	9.31E+03	1.01E+04	2.29E+03
$m=10$	2.64E+04	2.17E+04	1.17E+05	7.30E+03	2.90E+04	2.32E+04	1.74E+05	1.22E+04	2.99E+04	2.17E+04	1.43E+05	6.14E+03
$m=20$	5.74E+04	2.73E+04	3.54E+05	1.21E+04	8.78E+04	5.52E+04	4.76E+05	1.19E+04	4.96E+04	4.69E+04	3.47E+05	9.71E+03
均值	2.99E+04	1.79E+04	1.59E+05	6.93E+03	4.11E+04	2.77E+04	2.19E+05	8.88E+03	2.87E+04	2.60E+04	1.67E+05	6.05E+03

图 9 展示了 MOGWO 与其他三种对比算法在工件数为 100、机器数为 20（序列相关准备时间为 SDST-10 下第③种零空闲约束场景）的不同工厂的非支配解结果。从图中可以看出，MOGWO 算法所得出的结果具有显著的优势；其次，MOGWO 算法的种群分布性显著优于三种对比算法；最后，本文所提出的 MOGWO 是一种兼顾收敛性和分布性的最佳算法。

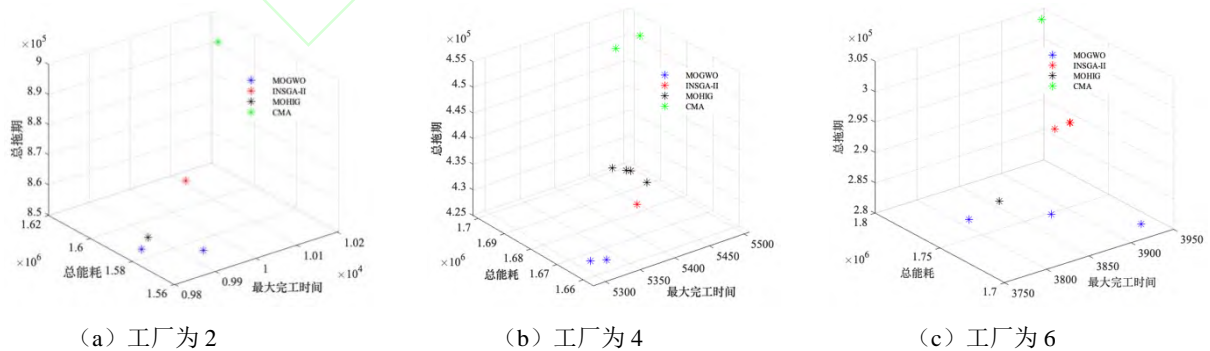
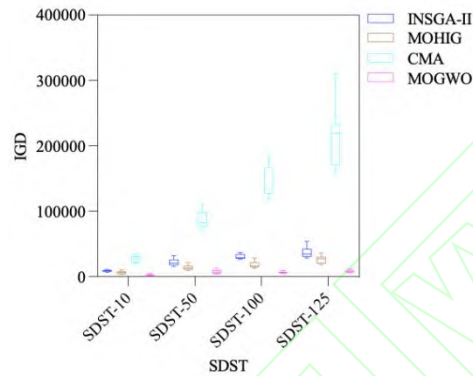


图 9 100 工件 20 机器在三个不同工厂的非支配解算法对比

通过对 MOGWO 与 INSGA-II、MOHIG 以及 CMA 在不同算例下的测试结果进行方差分析, 进一步验证各算法性能的差异。图 10 显示出四种算法的均值变化线以及 95% 置信度下的 Turkey's HSD 检验的置信区间。从图 10 可以发现, MOGWO 对于不同序列相关准备时间的算例所获得的 IGD 值均小于 INSGA-II、MOHIG 以及 CMA, 进一步说明, MOGWO 得到的计算结果质量更高且具有一



定的优越性。

图 10 MOGWO 与 INSGA-II、MOHIG、CMA 在不同 SDST 约束的方差分析

4.5 非参数统计检验

为了使比较结果更具有说服力, 对不同算法在表 13-16 中不同的序列相关准备时间的求解结果进行了统计分析, 并检验了不同算法之间的显著性差异。非参数统计分析方法使用了 Wilcoxon 符号秩检验^[38]与 Friedman 检验^[39], 显著性水平为 5%, 检验结果详见表 17-18。

表 17 各算法的 Wilcoxon 符号秩检验结果

SDST-10				SDST-50			
<i>F</i>	MOGWO VS.	<i>p</i> -value	<i>R</i>	<i>F</i>	MOGWO VS.	<i>p</i> -value	<i>R</i>
2	INSGA-II	1.63E-09	+	2	INSGA-II	8.34E-07	+
	MOHIG	1.83E-06	+		MOHIG	1.23E-06	+
	CMA	1.16E-09	+		CMA	2.33E-10	+
4	INSGA-II	1.63E-09	+	4	INSGA-II	7.68E-09	+
	MOHIG	4.81E-05	+		MOHIG	8.41E-06	+
	CMA	2.33E-10	+		CMA	8.34E-07	+
6	INSGA-II	2.33E-09	+	6	INSGA-II	2.33E-10	+
	MOHIG	1.58E-05	+		MOHIG	4.47E-06	+
	CMA	4.66E-10	+		CMA	2.33E-10	+
SDST-100				SDST-125			
<i>F</i>	MOGWO VS.	<i>p</i> -value	<i>R</i>	<i>F</i>	MOGWO VS.	<i>p</i> -value	<i>R</i>
2	INSGA-II	3.37E-06	+	2	INSGA-II	2.33E-10	+

	MOHIG	1.85E-05	+		MOHIG	8.34E-07	+
	CMA	6.99E-10	+		CMA	2.33E-10	+
4	INSGA-II	4.74E-07	+	4	INSGA-II	1.26E-05	+
	MOHIG	2.02E-05	+		MOHIG	5.77E-06	+
	CMA	4.66E-10	+		CMA	2.33E-10	+
6	INSGA-II	3.26E-09	+	6	INSGA-II	6.44E-07	+
	MOHIG	1.36E-06	+		MOHIG	1.30E-05	+
	CMA	1.11E-06	+		CMA	2.33E-10	+

表 17 展示了在不同序列相关准备时间的所有基准测试算例中 INSGA-II、MOHIG 以及 CMA 共三种算法分别与 MOGWO 的 Wilcoxon 符号秩检验中计算的 p 值, R 代表 MOGWO 的性能是否优于对比算法, 如果优于对比算法, 用“+”表示, 如果近似于对比算法, 用“=”表示, 否则, 用“-”表示。从表 17 结果可知, 所有测试算例的 p 值远小于 5%, 说明 MOGWO 与其他三种对比算法之间存在显著差异, 且 MOGWO 显著更优。

表 18 各算法的 Friedman 检验结果

SDST-10				SDST-50			
F	算法	秩均值	排名	F	算法	秩均值	排名
2	INSGA-II	2.70	2	2	INSGA-II	3.05	3
	MOHIG	2.89	3		MOHIG	2.55	2
	CMA	3.30	4		CMA	3.36	4
	MOGWO	1.11	1		MOGWO	1.05	1
	Chi-Square	55.59			Chi-Square	63.17	
	p -value	5.15E-12			p -value	1.24E-13	
4	INSGA-II	2.67	3	4	INSGA-II	2.88	3
	MOHIG	2.47	2		MOHIG	2.55	2
	CMA	3.70	4		CMA	3.44	4
	MOGWO	1.17	1		MOGWO	1.14	1
	Chi-Square	65.12			Chi-Square	58.05	
	p -value	4.72E-14			p -value	1.53E-12	
6	INSGA-II	2.76	3	6	INSGA-II	2.70	3
	MOHIG	2.23	2		MOHIG	2.50	2
	CMA	3.76	4		CMA	3.70	4
	MOGWO	1.26	1		MOGWO	1.11	1
	Chi-Square	66.07			Chi-Square	68.65	
	p -value	2.97E-14			p -value	8.31E-15	
SDST-100				SDST-125			
F	算法	秩均值	排名	F	算法	秩均值	排名

2	INSGA-II	2.86	3	2	INSGA-II	2.79	3
	MOHIG	2.24	2		MOHIG	2.76	2
	CMA	3.70	4		CMA	3.44	4
	MOGWO	1.20	1		MOGWO	1.02	1
	Chi-Square	67.14			Chi-Square	64.47	
	<i>p</i> -value	1.75E-14			<i>p</i> -value	6.50E-14	
4	INSGA-II	2.67	2	4	INSGA-II	2.74	3
	MOHIG	2.71	3		MOHIG	2.68	2
	CMA	3.41	4		CMA	3.48	4
	MOGWO	1.21	1		MOGWO	1.09	1
	Chi-Square	51.27			Chi-Square	60.70	
	<i>p</i> -value	4.29E-11			<i>p</i> -value	4.16E-13	
6	INSGA-II	2.82	3	6	INSGA-II	2.58	2
	MOHIG	2.48	2		MOHIG	2.59	3
	CMA	3.47	4		CMA	3.73	4
	MOGWO	1.23	1		MOGWO	1.11	1
	Chi-Square	53.18			Chi-Square	69.20	
	<i>p</i> -value	1.68E-11			<i>p</i> -value	6.33E-15	

表 18 数据结果可知，在 SDST-10 约束下，工厂数为 2、4、6 时，MOGWO 所求解出的秩均值分别为 1.11、1.17、1.26；在 SDST-50 约束下，工厂数为 2、4、6 时，MOGWO 所求解出的秩均值分别为 1.05、1.14、1.11；在 SDST-100 约束下，工厂数为 2、4、6 时，MOGWO 所求解出的秩均值分别为 1.20、1.21、1.23；在 SDST-125 约束下，工厂数为 2、4、6 时，MOGWO 所求解出的秩均值分别为 1.02、1.09、1.11，以上在不同 SDST 约束以及在不同工厂的秩均值远小于其他三种对比算法 INSGA-II、MOHIG 以及 CMA，因此，再次证明 MOGWO 具有良好的性能。另外，不同 SDST 在不同工厂的 Friedman 检验的 *p* 值均远小于 5%，说明 MOGWO 与其他三种算法之间存在显著的差异性。

5 结束语

针对 DMNIPFSP/SDST 的特征，一种多目标灰狼优化算法（MOGWO）被提出。首先，为提高初始种群的质量，设计了混合初始化策略；其次，在位置更新阶段，针对基本灰狼优化算法无法直接应用于求解离散车间调度问题的不足，提出了一种离散化种群更新机制，使算法能够直接在离散调度空间中进行搜索，并且设计了种群协同机制平衡三个优化目标；同时，针对不同的优化目标各设计了四种邻域策略进行局部搜索，有利于提升算法跳出局部最优的能力以及提高算法的求解精度。通过对四种不同序列相关准备时间场景下的三种不同类型零空闲约束条件的算例进行仿真实验，并将其与 INSGA-II、MOHIG 以及 CMA 进行对比，结果表明，MOGWO 在求解考虑序列相关准备时

间的多目标分布式混合零空闲置换流水车间调度问题具有显著的优势。下一步研究将多目标灰狼优化算法应用于其他约束条件的动态调度问题以及以多目标灰狼优化算法为基础设计更有效的求解算法。

参考文献:

- [1] FERNANDEZ-VIAGAS V, MOLINA-PARIENTE J M, FRAMINAN J M. Generalised accelerations for insertion-based heuristics in permutation flowshop scheduling[J]. *European Journal of Operational Research*, 2020, 282(3): 858-872.
- [2] BAGHERI RAD N, BEHNAMIAN J. Recent trends in distributed production network scheduling problem[J]. *Artificial Intelligence Review*, 2022, 55(4): 2945-2995.
- [3] NADERI B, RUIZ R. The distributed permutation flowshop scheduling problem[J]. *Computers & Operations Research*, 2010, 37(4): 754-768.
- [4] HUANG J P, PAN Q K, GAO L. An effective iterated greedy method for the distributed permutation flowshop scheduling problem with sequence-dependent setup times[J]. *Swarm and Evolutionary Computation*, 2020, 59: 100742.
- [5] ALMEIDA F S D, NAGANO M S. Heuristics to optimize total completion time subject to makespan in no-wait flow shops with sequence-dependent setup times[J]. *Journal of the Operational Research Society*, 2023, 74(1): 362-373.
- [6] ABREU L R, NAGANO M S. A new hybridization of adaptive large neighborhood search with constraint programming for open shop scheduling with sequence-dependent setup times[J]. *Computers & Industrial Engineering*, 2022, 168: 108128.
- [7] GUO H W, SANG H Y, ZHANG B, et al. An effective metaheuristic with a differential flight strategy for the distributed permutation flowshop scheduling problem with sequence-dependent setup times[J]. *Knowledge-Based Systems*, 2022, 242: 108328.
- [8] MIYATA H H, NAGANO M S. An iterated greedy algorithm for distributed blocking flow shop with setup times and maintenance operations to minimize makespan[J]. *Computers & Industrial Engineering*, 2022, 171: 108366.
- [9] KARABULUT K, ÖZTOP H, KIZILAY D, et al. An evolution strategy approach for the distributed

- permutation flowshop scheduling problem with sequence-dependent setup times[J]. *Computers & Operations Research*, 2022, 142: 105733.
- [10] LU C, ZHANG B, GAO L, et al. A knowledge-based multiobjective memetic algorithm for green job shop scheduling with variable machining speeds[J]. *IEEE Systems Journal*, 2021, 16(1): 844-855.
- [11] VALLADA E, RUIZ R, MINELLA G. Minimising total tardiness in the m-machine flowshop problem: A review and evaluation of heuristics and metaheuristics[J]. *Computers & Operations Research*, 2008, 35(4): 1350-1373.
- [12] CHENG C Y, YING K C, CHEN H H, et al. Minimising makespan in distributed mixed no-idle flowshops[J]. *International Journal of Production Research*, 2019, 57(1): 48-60.
- [13] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69: 46-61.
- [14] LU C, GAO L, PAN Q, et al. A multi-objective cellular grey wolf optimizer for hybrid flowshop scheduling problem considering noise pollution[J]. *Applied Soft Computing*, 2019, 75: 728-749.
- [15] LEI Bin, JIN Yantong, LIU Hailong. Job scheduling of cross layer shuttle-based storage system combined with FJSP problem[J]. *Computer Integrated Manufacturing Systems*, 2023, 29(8): 2751-2760 (in Chinese). [雷斌,金彦彤,刘海龙.结合 FJSP 问题的跨层穿梭车仓储系统作业调度[J]. *计算机集成制造系统*, 2023, 29(8): 2751-2760.]
- [16] TANG Hongtao, LI Yue, WANG Lei. An improved GWO algorithm for fuzzy distributed flexible job shop scheduling problem[J]. *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, 2022, 50(6): 81-88 (in Chinese). [唐红涛,李悦,王磊.模糊分布式柔性作业车间调度问题的求解算法[J]. *华中科技大学学报(自然科学版)*, 2022, 50(6): 81-88.]
- [17] GU Jiuchun, JIANG Tianhua, ZHU Huiqi. Energy-saving job shop scheduling problem with multi-objective discrete grey wolf optimization algorithm[J]. *Computer Integrated Manufacturing Systems*, 2021, 27(8): 2295-2306 (in Chinese). [顾九春,姜天华,朱惠琦.多目标离散灰狼优化算法求解作业车间节能调度问题[J]. *计算机集成制造系统*, 2021, 27(8): 2295-2306.]
- [18] LI X, XIE J, MA Q, et al. Improved gray wolf optimizer for distributed flexible job shop scheduling problem[J]. *Science China Technological Sciences*, 2022, 65(9): 2105-2115.
- [19] WANG C, ZHAO L, LI X, et al. An improved grey wolf optimizer for welding shop inverse

- scheduling[J]. Computers & Industrial Engineering, 2022, 163: 107809.
- [20] ZHU Yunhai, LUO Lufeng, LU Qinghua, et al. Flexible job shop dynamic scheduling method for order disturbance[J/OL]. Computer Integrated Manufacturing Systems: 1-23, (in Chinese). <http://kns.cnki.net/kcms/detail/11.5946.TP.20230223.1654.036.html>. [朱贇海,罗陆锋,卢清华,等. 面向订单扰动的柔性作业车间动态调度方法[J/OL]. 计算机集成制造系统:1-23,[2023-04-23]. <http://kns.cnki.net/kcms/detail/11.5946.TP.20230223.1654.036.html>.]
- [21] MANSOURI S A, AKTAS E, BESIKCI U. Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption[J]. European Journal of Operational Research, 2016, 248(3): 772-788.
- [22] PAN Q K, GAO L, LI X Y, et al. Effective constructive heuristics and meta-heuristics for the distributed assembly permutation flowshop scheduling problem[J]. Applied Soft Computing, 2019, 81: 105492.
- [23] LIU J, REEVES C R. Constructive and composite heuristic solutions to the $P/\sum C_i$ scheduling problem[J]. European Journal of Operational Research, 2001, 132(2): 439-452.
- [24] WANG H, WANG W, SUN H, et al. A new cuckoo search algorithm with hybrid strategies for flow shop scheduling problems[J]. Soft Computing, 2017, 21: 4297-4307.
- [25] ABDEL-BASSET M, MANOGARAN G, EL-SHAHAT D, et al. A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem[J]. Future Generation Computer Systems, 2018, 85(1): 129-145.
- [26] LI X, YIN M. A hybrid cuckoo search via Lévy flights for the permutation flow shop scheduling problem[J]. International Journal of Production Research, 2013, 51(16): 4732-4754.
- [27] LU C, GAO L, LI X Y, et al. A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry[J]. Engineering Applications of Artificial Intelligence, 2017, 57: 61-79.
- [28] ZHAO F, HU X, WANG L, et al. A memetic discrete differential evolution algorithm for the distributed permutation flow shop scheduling problem[J]. Complex & Intelligent Systems, 2022, 8(1): 141-161.
- [29] DENG J, WANG L. A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem[J]. Swarm and Evolutionary Computation, 2017, 32: 121-131.

- [30] ALLALI K, AQIL S, BELABID J. Distributed no-wait flow shop problem with sequence dependent setup time: Optimization of makespan and maximum tardiness[J]. Simulation Modelling Practice and Theory, 2022, 116: 102455.
- [31] 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.
- [32] RUIZ R, MAROTO C, ALCARAZ J. Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics[J]. European Journal of Operational Research, 2005, 165(1): 34-54.
- [33] RUIZ R, STUTZLE T. An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives[J]. European Journal of Operational Research, 2008, 187(3): 1143-1159.
- [34] LEI D, SU B. A multi-class teaching-learning-based optimization for multi-objective distributed hybrid flow shop scheduling[J]. Knowledge-Based Systems, 2023, 263: 110252.
- [35] ZENG Q Q, LI J Q, LI R H, et al. Improved NSGA-II for energy-efficient distributed no-wait flow-shop with sequence-dependent setup time[J]. Complex & Intelligent Systems, 2023, 9(1): 825-849.
- [36] LU C, LIU Q, ZHANG B, et al. A pareto-based hybrid iterated greedy algorithm for energy-efficient scheduling of distributed hybrid flowshop[J]. Expert Systems with Applications, 2022, 204: 117555.
- [37] MONTGOMERY D C. Design and analysis of experiments[M]. John wiley & sons, 2017.
- [38] WILCOXON F. Individual comparisons by ranking methods[C]. Springer, New York, 1992: 196-202.
- [39] MEDDIS R. Unified analysis of variance by ranks[J]. British Journal of Mathematical and Statistical Psychology, 1980, 33(1): 84-98.

作者简介:

陈水琳（1995-），女，江西南昌人，博士研究生，研究方向：算法优化与车间调度，E-mail: 1219088@mail.dhu.edu.cn;

+郑建国（1962-），男，福建龙岩人，教授，博士生导师，研究方向：智能决策与数据挖掘、技术经济分析、进化计算与优化算法等，通讯作者，E-mail: 13767425531@163.com。

