

分类号 F27
收藏编号
学校代码 10386



密级 公开
学号 180727066
编号

福州大学

工程硕士专业学位研究生学位（毕业）论文
(应用研究)

考虑序列依赖准备时间的多目标并行机作业
车间调度问题研究

工程领域： 工业工程
研究方向： 生产与服务系统工程
研究生姓名： 沈阳
指导教师、职称： 刘仕强教授
协助导师、职称： 李德彪副教授
所在学院： 经济与管理学院
答辩委员会主席：

二〇二一年六月

考虑序列依赖准备时间的多目标并行机作业车间调度问题研究

摘要

车间调度是制造企业生产管理的核心，决定了生产资源能否合理安排。传统的车间调度问题没有考虑序列依赖准备时间，而在柔性制造系统中需要加工多品种、不同规模的工件，序列依赖准备时间真实存在。实际生产中，企业通常需要对多个目标同时决策，且产品的制成需要机器和其他多种辅助资源，而辅助资源的限制会对生产造成显著影响。因此，在研究带有序列依赖准备时间的多目标并行机作业车间调度问题时考虑资源约束具有重要的现实意义。

首先，本文研究了不同工件在同一机器上的序列依赖准备时间，选择最小化总完工时间、最小化总延迟时间和最小化总准备时间作为调度目标，建立了具有序列依赖准备时间的多目标并行机作业车间调度模型。将快速非支配排序遗传算法（NSGA2）作为算法框架，将人工蜂群算法（ABC）修改为离散型多目标人工蜂群算法（MOABC）并与 NSGA2 融合，提出了混合 NSGA2。使用分段编码方式，并根据染色体结构特点，采用 IPOX 和 RPX 混合交叉操作。NSGA2 产生子代种群后使用采蜜蜂和跟随蜂对子代种群中的个体进行邻域搜索，通过贪婪选择方式选出新个体；侦查蜂随机产生新个体替换被舍弃的个体。蜂群搜索结束后产生的新子代种群与 NSGA2 产生的父代种群合并，通过精英策略筛选出新的父代种群，求出 Pareto 最优解。运用混合 NSGA2 求解相关算例，通过与其他算法在多目标优化评价指标上的比较，验证了算法的有效性。

其次，本文研究了相同工件在不同机器上转换时产生的序列依赖准备时间，增加了每个工序需要多种辅助资源的限制条件。建立了具有多资源约束和序列依赖准备时间的多目标并行机作业车间调度问题数学模型，提出了两阶段解码策略。建立加工资源信息集合，以保存各时刻所有加工资源的信息。通过资源信息的更新，按照工件等待开始加工时间最短原则确定各个资源的分配方式和各工序的开始加工时间。使用大规模的算例测试了带有两阶段解码策略的混合 NSGA2，通过多目标优化评价指标结果表明了混合 NSGA2 在算法支配性、解的多样性以及解的质量上均优于其他对比算法。

关键词：序列依赖准备时间；并行机作业车间；混合快速非支配排序遗传算法；多资源约束；多目标优化

Study on multi-objective parallel machine job shop scheduling problem with sequence-dependent setup time

Abstract

Job shop scheduling is the core of manufacturing enterprise production management, which determines whether the production resources are reasonably arranged or not. Traditional job shop scheduling issues do not consider sequence-dependent setup time, and only one resource of the machine is taken into account. In the flexible manufacturing system, it is necessary to process multiple varieties and different batches of jobs, and the sequence-dependent setup time really exists. In actual production, companies usually need to make decision on multiple objectives, and the process of production requires machines and other auxiliary resources, while the auxiliary resource restrictions may have a significant impact on production. Therefore, it is important to consider resource constraints when studying a multi-objective parallel job shop scheduling problem with sequence-dependent setup time.

Firstly, this paper studies the sequence-dependent setup time of different jobs on the same machine, chooses the makespan, minimizing total delay time and minimizing total setup time as the scheduling objectives and establishes the multi-objective parallel machine job shop scheduling with the sequence-dependent setup time model. A fast non-dominated sorting genetic algorithm (NSGA2) is used as an algorithm frame, and the Artificial Bee Colony (ABC) algorithm is modified to discrete Multi-objective Artificial Bee Colony (MOABC) algorithm combined with NSGA2. The segmentation coding method is used. Two types of crossover operation, IPOX and RPX, are used according to the structural characteristics of the chromosome. After NSGA2 generates the progeny population, the employed bees and onlooker bees in the MOABC algorithm part use two neighborhood search methods to optimize the progeny population according to the pareto dominance. The bees select a new individual through the greed selection method. If one individual is abandoned, the scout bee is used to generate a new one. The new formed progeny population is merged with the paternal generated by NSGA2 to find the pareto optimal solution through the elite strategy. The hybrid NSGA2 is used to solve related calculation examples, and compared with other

algorithms using multi-objective optimization evaluation indicators, which verifies the effectiveness of the algorithm.

Secondly, this paper studies the sequence-dependent setup time generated when the same job is converted on different machines, and the restriction conditions that each process requires multiple auxiliary resources are added. A mathematical model for multi-resource constraint multi-objective parallel machine job shop scheduling problem with sequence dependent setup time is established, and two-stage decoding method is proposed. A set is established to save all processing resource information. Through the update of resource information, the allocation mode of each resource and the start processing time of each operation is determined according to the shortest principle of job interval time. The hybrid NSGA2 with a two-stage decoding strategy is tested through some large-scale examples. The results of multi-objective optimization evaluation indicators show that the hybrid NSGA2 is superior to other comparison algorithms in terms of algorithm dominance, pareto solution quality and pareto solution diversity.

Key words: sequence-dependent setup time; parallel machine job shop; hybrid fast non-dominated sorting genetic algorithm; multi-resource constrained; multi-objective optimal

目 录

第一章 绪论.....	1
1.1 研究背景及意义.....	1
1.1.1 研究背景.....	1
1.1.2 研究意义.....	1
1.2 国内外研究现状分析.....	3
1.2.1 并行机作业车间调度问题概述.....	3
1.2.2 并行机作业车间调度问题研究方法及现状	3
1.2.3 序列依赖准备时间车间调度问题研究方法及现状	5
1.2.4 多资源约束车间调度问题研究方法及现状	6
1.2.5 NSGA2 与 ABC 算法.....	7
1.3 研究现状总结与问题分析	8
1.4 研究内容及目标.....	9
1.5 主要的创新点.....	10
第二章 相关问题描述与理论基础	12
2.1 引言.....	12
2.2 考虑准备时间的车间调度问题分类	12
2.2.1 按照准备时间与序列相关性分类.....	12
2.2.2 按照考虑准备时间调度问题目标分类	13
2.2.3 按照工序排序和机器转换的准备时间分类	14
2.3 SDST-PMJSS 问题描述	15
2.4 快速非支配排序遗传算法(NSGA2).....	16
2.4.1 快速非支配排序.....	17
2.4.2 拥挤距离.....	17
2.4.3 拥挤度比较算子.....	18
2.4.4 精英策略.....	18
2.4.5 NSGA2 的特点	18
2.5 人工蜂群算法(ABC).....	19
2.5.1 ABC 算法相关参数定义及流程.....	19
2.5.2 算法流程.....	21
2.5.3 ABC 算法的特点	23
2.6 多目标优化.....	24
2.6.1 多目标优化问题基本理论.....	24
2.6.2 多目标优化评价指标.....	25
2.7 本章小结.....	26

第三章 基于混合 NSGA2 的多目标 SDST-PMJSS 问题研究.....	27
3.1 引言.....	27
3.2 多目标 SDST-PMJSS 问题模型构建.....	27
3.2.1 问题假设.....	27
3.2.2 相关变量和参数定义.....	27
3.2.3 数学模型构建.....	28
3.3 混合 NSGA2 求解多目标 SDST-PMJSS 问题.....	30
3.3.1 混合原理.....	30
3.3.2 种群初始化.....	31
3.3.3 个体的编码.....	31
3.3.4 解码.....	32
3.3.5 交叉操作.....	33
3.3.6 变异操作.....	34
3.3.7 外部存档的更新.....	35
3.3.8 多目标 ABC 算法.....	36
3.3.9 混合 NSGA2 算法步骤.....	36
3.4 实验设计与结果分析.....	39
3.4.1 仿真环境.....	39
3.4.2 实验数据设计.....	39
3.4.3 参数设置.....	39
3.4.4 实验结果与分析.....	40
3.5 本章小结.....	44
第四章 多资源约束多目标 SDST-PMJSS 问题研究.....	45
4.1 引言.....	45
4.2 问题描述.....	45
4.3 多资源约束多目标 SDST-PMJSS 问题模型构建.....	47
4.3.1 相关变量和参数定义.....	47
4.3.2 数学模型构建.....	47
4.4 混合 NSGA2 求解多资源约束多目标 SDST-PMJSS 问题.....	49
4.4.1 种群初始化.....	49
4.4.2 个体的编码.....	49
4.4.3 两阶段解码策略.....	50
4.5 实验设计与结果分析.....	54
4.5.1 仿真环境.....	54
4.5.2 实验数据设计.....	54
4.5.3 参数设置.....	55
4.5.4 实验结果与分析.....	55
4.6 本章小结.....	60

研究总结与展望	61
(一)研究总结	61
(二)研究展望	62
参考文献.....	63

第一章 绪论

1.1 研究背景及意义

1.1.1 研究背景

制造业是国民经济的命脉，是国家硬实力的象征。2020 年 1 月中国智能制造系统解决方案大会提出，我国要坚持智能制造，提高系统解决方案供给能力，以推动制造业高质量发展、加快制造强国建设。在此背景下，我国需要实施制造强国战略，促使制造业迅速向信息化、智能化改进。

在产业化升级的进程中，企业需要扩大优势提高产品质量，降低成本。企业若想得到长期的良性发展，需要在如今瞬息万变的市场环境中提高产品质量和生产效率以获得更多的市场占有率。如此，企业有诸多问题需要考虑：如何根据市场需求选择调度方案，如何提高企业高效管理的能力。

调度理论是运筹学组合优化领域中一个非常重要并且活跃的分支，在学者对调度问题几十年的研究中已产生了丰厚的成果。调度问题的模型背景大多来源于实际，最开始调度技术主要应用于机器制造领域，随着其成果和理论日益发展，应用领域扩大到了生产计划调度、物流供应链管理、计算机科学、航空、铁路、医疗、大数据等。对调度问题的研究从简单到复杂，从原先的单机调度（Single Machine Scheduling）发展到并行机调度（Parallel Machine Scheduling）；从流水车间（Flow-Shop）到作业车间（Job-Shop）、自由车间（Open-Shop）等^[1]。调度问题从实际中产生，调度模型以及相关算法需要贴合实际需求，每一种调度问题的解决都可以在实际中找到例子，带有序列依赖准备时间的并行机作业车间（Sequence Dependent Setup Times Parallel Machine Job Shop, SDST-PMJSS）调度问题应运而生。在生产过程中，企业通常是多个部门协同生产，不同部门有自己的要求，因此实际中，为了避免各目标冲突，需要综合考虑调度计划，将多目标与 SDST-PMJSS 同时考虑是非常有现实意义的。

在复杂的生产过程中，每一个工序都需要相应的辅助资源，而这些辅助资源通常有限，无法同时满足加工要求，因此在该问题基础上增加多资源约束这一条件并加以求解在智能车间生产过程中更具有实际价值。

1.1.2 研究意义

1979 年 Graham^[2]等提出了三参数法 $\alpha / \beta / \gamma$ 来描述车间调度问题。参数 α

用以描述机器环境, 参数 β 用以描述加工工件的特征, 参数 γ 用以描述目标函数。

作业车间调度问题通常定义为 $J \parallel C_{max}$, 也称为异序作业车间, 在作业车间中每个工件都有自己预定的工艺路线, 机器的加工顺序和流水作业车间不同, 不是按照同一顺序进行加工。并行机作业车间 (Parallel Machine Job Shop, PMJSS) 定义为 $PJ \parallel C_{max}$, 作为作业车间 (Job-Shop) 的延伸, 已经被证明为 NP-hard 问题。多并行机作业车间调度问题的机器环境相对于作业车间环境或并行机器环境更为复杂, 从整体上看它是并行机, 而在并行机器环境内部又是作业车间环境, 所以可以认为该机器环境为作业车间和并行机器的一种混合。并行机作业车间调度问题的基本模型可以描述为: 在一个加工系统中有 n 个工件, 每个工件有 h 道工序, 所有工件需要按照工艺流程在 k 台机器上进行加工, 相应的工序可以在并行机器单元中选择一台机器进行加工, 要求确定机器上各工序排序顺序, 满足优化目标。

目前, 并行机作业车间广泛应用在柔性制造系统中。企业生产通常会使用一组并行机器, 以提高生产率, 并在发生机器故障时避免生产线停止运转, 多并行机作业车间更为符合现在众多的制造型企业, 车间可以根据自身情况分布单机和并行机器用以满足相应的制造目标要求。例如, 在显示器电子制造厂, 一台显示器有多种工件, 有些工件工艺简单或需求量比较小可以在一台固定的机器上完成加工; 而有些工件工艺复杂或需求量比较大, 为了使上游机器不产生堵塞和下游机器操作时间均匀, 需要设立并行机器, 以降低线平衡损失率。在工业生产中, 与顺序相关的生产准备时间普遍存在, 如加工过程中磨具替换; 布料染色过程中, 从深色到浅色, 设备需要清洗, 因此准备时间就比较长。然而, Kopanos^[3]等指出 90% 的调度研究都忽视了生产准备时间的存在。在一般并行机作业车间调度问题的研究中, 准备时间通常不被考虑, 或者准备时间与序列不相关。准备时间不是一个可有可无的问题, 为提高生产效率, 降低成本, 在模型建立与求解中必须将准备时间考虑在内。

在现代制造企业, 多品种的加工方式决定了制造系统必须具有高柔性, 但是通常在加工过程中产品制成不仅需要相关机器, 更需要额外的资源辅助加工。辅助资源通常有限, 资源的限制会对生产造成显著影响, 成为一个重要的实际问题。同时, 在文献检索中发现多资源约束这一问题并没有受到广泛关注, 因此本文的研究在工程和理论上都具有一定的意义。

1.2 国内外研究现状分析

1.2.1 并行机作业车间调度问题概述

车间调度问题主要目标是寻找最优的生产资源分配方法,通过满足某些约束条件来实现所追求的生产性能指标最优。1954 年,Johnson^[4]在研究具有两台机床的流水车间(Flow-Shop)调度问题后提出了 $n/2/F/C_{max}$ 和部分比较特殊的 $n/3/F/C_{max}$ 问题的求解方法,此后被学术界认为这是经典调度理论诞生的重要标志。随着制造技术的进步,调度理论也随之发展,多并行作业车间调度问题可以看成并行机车间调度问题和作业车间调度问题的混合。

(1) 并行机车间调度 (Parallel machine scheduling)

并行机车间调度问题的设备环境是在每个加工阶段至少有一台机器并联,构成并行机器单元。在生产过程中,每个工件可以选择并行机器单元中的一台机器完成加工^[5]。调度任务是按照工艺流程要求给各工件分配加工机器,并且确定每台机器上各工序的加工顺序,以达到优化目标要求。根据机器特点可进一步分为同型并行机 (Identical machines in parallel)、不相关并行机 (Unrelated machines in parallel)、异构并行机 (Machines in parallel with different speeds)。在同型并行机环境中,每个工件在任意机器上的加工时间以及相邻工件之间所需要的设置时间都是相同的^[6]。在异构并行机环境中每台设备加工同一工件的速度不一样,不同机器之间处理相邻工件所需要的设置时间都不完全相同^[7]。在不相关并行机环境中,每个阶段至少存在一台机器可用于加工,且整个加工过程中至少有一个阶段存在并行机器单元,各机器加工同一工件时间不同,互不相关,取决于工件与机器的匹配程度^[8]。

(2) 作业车间 (Job-shop)

在传统的作业车间调度问题中,各工件都有自己的工艺流程,每个工件有多道工序需要在特定的机器上加工。因此,在作业车间中每个工序能选择的加工机器数等于 1,在调度过程中需要按照工艺要求确定每台机器上各工序的加工顺序,以达到优化目标^[9]。

多并行机作业车间调度问题在高柔性现代制造系统中比较普遍,同型并行机由于并行机环境中加工时间相同,因此可以看成是柔性作业车间的一个特例^[10]。而同一工件在并行机环境中加工时间不同,即异构并行机,同样具有机器柔性,则该问题也可以称为柔性作业车间调度问题^[11]。

1.2.2 并行机作业车间调度问题研究方法及现状

作业车间调度问题属于组合优化问题,通过调度方案来合理分配资源,以提

高企业生产效率、质量和柔性，具有很强的现实意义。国内外很多学者都对车间调度问题进行研究，主要集中在建模和算法研究。按照算法精度，求解作业车间调度问题的算法主要分为精确算法和近似算法。

精确算法是保证能求出最优解的算法，是学者们研究的一类具有代表性的经典算法。分支定界算法（Branch and Bound method）、拉格朗日法（Lagrangian Relaxation）、动态规划法（Dynamic Programming）等是目前比较有代表性的精确算法。在上世纪许多学者进行了研究，将分支定界法运用到车间调度问题。Pinson^[12]提出分支定界算法成功对 10*10 作业车间调度问题进行求解，解决了近 25 年来无法求解的问题。Oezgüven^[13]等采用混合整数规划模型求解了准备时间发生在不同工件之间的柔性作业车间调度问题。Hansmann^[14]等将混合整数规划模型与分支定界算法相结合，并应用于铁路堵塞调度问题。Chen 和 Luh^[15]将最小化最大完工时间作为调度目标，利用动态规划法来求解作业车间调度问题，将动态规划法和拉格朗日松弛法通过实验进行比较，证明了前者效率更高。

精确算法在小规模问题上相对比较适用，能够求出最优解。随着工业水平的发展，制造型企业的规模越来越大，制造工艺越来越复杂，而这些大规模复杂的车间调度问题需要很快的算法求解速度以应对苛刻的市场环境，精确算法很难应用到实际的生产调度中去。鉴于精确算法的局限性，而近似算法在合理的时间范围能得到近似最优解，受到学者广泛关注和研究。

近似算法分为构造启发式算法和智能优化算法，为了增强启发式算法的优化能力，在基于调度规则的基础上许多学者提出了复杂的启发式算法。Adams^[16]提出了移动瓶颈算法对作业车间调度问题进行求解，将问题分解为若干单机子问题，依次确定每台机器上工件的加工顺序来获得最优解，该方法虽然能获得较好的调度方案，但计算时间过长。Liu 和 Kozan^[17]将多并行机作业车间调度问题分为单机调度和并行机调度，通过移动瓶颈法进行求解。

智能优化算法目前是调度优化领域中最常用的一种方法，能提高求解质量和效率，为解决许多实际问题提供一些新的思路^[18]。目前，比较有代表性的智能优化算法有：遗传算法（Genetic Algorithm, GA）、模拟退火算法（Simulate Anneal, SA）、禁忌搜索算法（Tabu Search, TS）、蚁群算法（Ant Colony Optimization, ACO）、人工蜂群算法（Artificial Bee Colony, ABC）、粒子群算法（Particle Swarm Optimization, PSO）、差分进化算法（Differential Evolution Algorithm, DEA）等。单一的智能优化算法均有其优缺点，因此学者们在求解多并行机车间调度问题时将多种智能算法混合来高效、高质量求解。Suresh^[19]等在研究相同并行机车间调度问题提出了邻域可变的搜索算法，并且和 SA 进行比较，通过大量数据实验说明了基于邻域可变的搜索算法在方案质量和一致性方面始终优于 SA。Chaudhry

和 Elbadawi^[20]以总拖延时间最小为目标为相同并行机调度问题提出一种通用的遗传算法,可自定义且易于实现。Lausch 和 Mönch^[21]提出了一种新的遗传算法、蚁群算法和大型邻域搜索算法,基于随机生成的相同并行机批调度问题比较了三种算法,说明了大型邻域搜索算法优于其他两种算法。Rajkanth^[22]等将相同并行机调度问题分成工件分配和工件排序两个阶段,提出了混合多机遗传算法进行求解。Hamzadayi 和 Yildiz^[23]解决了具有相同服务器和序列相关设置时间的静态多并行机的调度问题,建立了混合整数规划模型,开发了混合 SA-GA 算法。Aalaei^[24]等针对允许抢占的情况下,以总加权提前和延误和待处理在制品的持有成本为目标采用 NSGA 和 NSGA2 来求解目标问题,然后使用三个测量因子来评估算法性能。Kayvanfar^[25]等研究了具有可控处理时间的并行机作业车间调度问题,采用智能水滴算法进行求解。Wang^[26]等考虑能耗问题,以总能耗和制造期为双目标,提出一种具有局部搜索策略的构造启发式方法,运用 NSGA2 成功求解了该问题。

1.2.3 序列依赖准备时间车间调度问题研究方法及现状

对序列依赖准备时间的调度问题研究最早起源于 20 世纪 60 年代,此后,车间调度技术大力发展,在近几年对该问题的理论和方法上研究有诸多成果。

在流水车间中,An^[27]等考虑了两台机器的流水车间调度问题,并且第二台机器调整时间与工序相关,开发了启发式算法通过随机实例验证了该算法在求解 30 个工件以下的有效性。Ince^[28]等在研究具有序列依赖准备时间的置换流水车间调度问题时,提出了离散型人工蜂群算法,通过实验验证算法的有效性。Harbaoui^[29]等使用遗传算法求解了具有专用机器与序列有关的设置时间的两阶段混合流水车间,并成功应用于面食生产行业。Ahonen 和 Alvarenga^[30]将该问题与可重入柔性流水车间一起考虑,其中某个阶段的作业处理时间可能取决于当前阶段之前和之后对作业所做的决策,也会取决于作业在处理流程中所采用的机器顺序,并提出了模拟退火和禁忌搜索两种混合元启发式方法。Zhang^[31]等提出了一种新的混合分布估计算法,用于求解阻塞并且具有顺序依赖准备时间的流水车间调度问题,并且开发了一种基于参考序列的具有路径重新链接技术的搜索算法。Li^[32]等通过学习遗忘效应构建了基于位置的学习和遗忘效应模型解决了顺序相关无等待流水车间问题。

在作业车间中,Azzouz^[33]等通过遗传算法解决了带有序列依赖准备时间的最小化最大完工时间和双准则目标函数的双目标柔性作业车间调度问题。Sharma 和 Jain^[34]等为随机的动态作业车间调度问题提供了新的面向安装程序的调度规则,同时考虑了与序列相关的安装时间,建立了随机动态作业车间制造系统的离散事件仿真模型。Gallego^[35]等考虑了待机调度上一组作业过程,该作业受任意发

布日期和序列相关设置时间的限制,成功地在最大程度减少所有作业的最大完工时间。Yue^[36]等提出了一种具有遗传算法若干步骤的新型混合 Pareto 人工蜂群算法,成功求解了与序列相关设置时间的组调度问题。Wang^[37]等研究了不相关并行机环境下序列依赖准备时间的调度问题,提出了一种带有迭代贪婪搜索的混合分配算法。Shen^[38]等以禁忌搜索算法为算法框架,开发了一种特殊的邻域搜索方式和算法结构,解决了具有 SDST 的柔性作业车间调度问题。Bektu 和 Sarac^[39]将该问题进行扩充,提出了一种混合整数线性规划模型和 TS、SA 求解了序列依赖准备时间和机器限制的不相关并行机调度问题。Kim^[40]等针对带有序列相关准备时间的相同并行机调度问题提出了一种具有元启发式的新颖数学模型,并提出了两种用于元启发式解决方案的编码方案,以及三种用于从元启发式解决方案中获取调度的解码方法。

国内对序列依赖准备时间的车间调度问题研究起步相对比较晚,胡大勇和姚振强^[41]开发了两段染色体表达的遗传算法,通过不同规模问题实例验证了该算法在求解调整时间与顺序相关的等同并行机的有效性。徐建有^[42]等设计了一种基于动态邻域的多目标变深度搜索算法,引入 Pareto 最优概念求解了序列依赖准备时间的多目标流水车间调度问题。李冰和黎展滔^[43]以陶瓷行业生产车间为背景,建立了准备时间与序列相关的能耗优化调度问题数学模型,提出了基于 NEH 算法的混合遗传算法。查靓^[44]等开发了两阶段遗传算法,算法第一阶段安排调整时间,第二阶段将维护任务概率插入,求解了带有设备维护约束和序列依赖准备时间的柔性作业车间集成调度问题。杨武成和程文明^[45]将优先关系约束融入到交叉变异算子中,开发了变邻域搜索算法,成功解决了生产单一模式下顺序相关调整时间的双边装配线问题。

1.2.4 多资源约束车间调度问题研究方法及现状

目前大部分关于多并行机作业车间调度问题只将机器作为一种加工资源,而实际生产中生产能力通常受多种辅助资源的限制,每一种工序所需的资源没有得到合理分配都会成为提高生产经济成本。在柔性制造系统中,工件有多道工序需要在不同的机器上完成加工,工件在机器上的切换机器需要一定的准备时间,因此将基于机器的序列依赖准备时间和多资源约束两个真实的限制条件考虑进去,使调度理论的应用更加贴合于真实的生产环境,对现今智能制造的发展具有重大意义。

Afzalirad 和 Rezaeian^[46]开发出了遗传算法和人工免疫算法求解了具有资源约束,序列依赖准备时间和工件有不同发布日期与优先级约束的不相关并行机调度问题,并应用于某造船厂区块安装工作中。Zhen 和 Wang^[47]研究了具有工人和

机器的双资源柔性作业车间调度问题,提出了一种基于知识导向的果蝇优化算法,将知识引导搜索和气味搜索结合在一起提高了算法的搜索能力。Mehdi^[48]等运用模拟退火算法和减振优化算法,利用多种邻域结构在解空间进行搜索,求解了带有工人约束的双资源柔性作业车间调度问题,并使用随机生成的测试问题评估所开发算法性能。Wang 和 Ming^[49]等在研究多资源约束多目标并行机调度问题同时考虑了资源预防性维护活动,提出了一种基于自适应性 NSGA2 的多目标集成优化方法。Villa^[50]等在不相关并行机问题基础上增加紧缺资源的限制条件,提出了两种启发式策略,一是基于整个解决方案对资源进行分配;二是先从不考虑资源约束的分配规则开始,再修复不可行的分配方案以获得可行解。Fu^[51]等研究了一种动态资源分配的并行机调度问题,提出了一种主从遗传算法来确定机器选择、工序排序和资源分配三个子问题,并采用贪婪启发式规则来确定作业的开始时间。Zhong^[52]等针对多资源约束多目标柔性作业车间调度问题,建立了以完工时间、人员分配、设备依从性和生产成本四个目标的数学模型,在 NSGA2 中引入多规则动态规划以获得最优完工时间和人员分配方案。Li^[53]等提出了一种分支种群遗传算法来最小化加工成本和完工时间。

国内学者近些年也有部分对多资源约束车间调度问题进行了研究,李兢尧^[54]等提出了压缩时窗调度策略和分支种群遗传算法,给出了优化成本和优化最小化总完工时间的压缩时窗调度条件,求解了双资源约束车间调度问题。孙斌^[55]等研究了基于工序的多资源约束车间调度问题,将差分进化算子和 Baldwinian 学习策略引入萤火虫算法中。陈少^[56]等将模拟退火算法引入鸟群算法,采用三条链组合编码方式提出改进鸟群算法求解双资源约束的车间调度问题。娄航宇^[57]等将资源约束柔性作业车间调度理论运用到航空结构件生产车间,提出多小组协同教与学优化算法。李俊青^[58]等考虑运输资源约束和工件间的准备时间,设计了一种机床分配邻域结构,在人工蜂群算法的三个阶段提出了不同的改进策略以提高算法的全局搜索能力。

1.2.5 NSGA2 与 ABC 算法

Deb^[59]等于 2002 年在 NSGA 算法基础上提出了快速非支配排序遗传算法(Fast Non-dominated Sorting Genetic Algorithm, NSGA2),目前 NSGA2 已广泛应用于多目标优化领域。在解决车间调度问题时,Chen^[60]等针对机器故障的柔性作业车间调度问题提出了一种包括转移重新安排策略和完整重新安排策略解决随机故障机器,并且采用 NSGA2 进行求解。Gong^[61]等建立了具有工人柔性的非线性整数模型,提出一种自适应遗传算子,消除了 NSGA2 算法中的重复个体。Xu^[62]等以 NSGA2 为框架,将文化基因算法与其融合,提出了基于循环拥挤距离的精

英策略,增加了人口分布的多样性。陈辅斌^[63]等引入免疫平衡原理改进了 NSGA2 选择策略和精英策略,增强了局部搜索能力。张守京和王彦亭^[64]等分析了柔性车间调度问题,将基于加工机器状态的竞争选择策略引入 NSGA2 中,设计了基于拥挤度的自适应交叉算子,改进了 NSGA2 容易早熟的缺点,提高了求解质量。

Karaboga^[65]于 2005 年系统地提出了人工蜂群算法 (Artificial Bee Colony Algorithm, ABC),之后众多学者对其进行研究。Sonmez^[66]指出人工蜂群算法作为一种相对比较新的群体优化的智能算法,由于其控制参数量比其他算法少,更易于实现且优化性能比较好引起了学者们的兴趣,已成功应用于车间调度优化问题。Pan^[67]在解决钢铁生产中出现的新的炼钢连铸并行机调度问题时,在人工蜂群算法中引入自适应邻域算子。Zhang^[68]等设计了树搜索算法增强 ABC 的开发能力,解决了以总加权拖延最小为目标的作业车间调度问题。Chang^[69]等设计了一种有效的多目标人工蜂群算法,成功求解服装厂染色过程中以延迟成本和染色缸利用率为双目标的并行机批处理问题,并通过实验证明所提算法均优于通用的多目标调度算法。Liu 和 Lei^[70]研究了分布式不相关并行机调度问题,提出了一种带有分区的人工蜂群算法。对于柔性作业车间调度问题,毕孝儒^[71]等给出以关键路径为导向的变异操作,将遗传算法与人工蜂群算法混合,求解了无等待多目标柔性车间调度问题。吴帅和姚兆波^[72]采用扩展的 NEH 算法产生初始蜂群,同时改进雇佣蜂邻域搜索方式和侦查蜂产生新解的方式提高算法的搜索效率,解决了带有启动时间的批量混合流水车间调度问题。郑小操和龚文引^[73]为提高初始种群的多样性,引入了混沌理论求解模糊柔性作业车间调度问题。李益兵^[74]等提出了一种集成式的编码和对应解码方案,同时为增强人工蜂群算法的局部搜索能力,在采蜜蜂搜索结束后引入了一种有效的动态邻域搜索操作,求解了多目标绿色柔性作业车间调度问题。

1.3 研究现状总结与问题分析

通过上述对国内外研究现状分析总结可以得出以下结论:

(1) 序列依赖准备时间的多目标并行机作业车间调度问题非常适用于现在的柔性制造系统,有着很强的应用背景。多目标问题是一个综合决策问题,符合实际制造企业的需求,因此具有重要实际研究意义。

(2) 多资源约束多目标序列依赖准备时间的多目标并行机作业车间的研究仍处于起步状态,但是实际加工过程中多资源约束的情况经常存在,制约着车间的生产能力。该问题复杂度更高,学者们通常考虑单目标且较少运用 NSGA2 或 ABC 算法进行求解。

(3) 随着智能制造水平的提高,问题的组合复杂度越来越大,近似算法是求

解考虑序列依赖准备时间的多目标并行机作业车间调度问题的主流方法。但是，已知的单一的算法各有优缺点也难以取得突破性进展。因此，开发混合算法，使用混合 NSGA2 来求解序列依赖准备时间多目标并行机作业车间调度问题也具有理论价值。

1.4 研究内容及目标

本文主要研究考虑序列依赖准备时间的多目标并行机作业车间调度问题，针对该问题，将数学和计算机技术结合，对车间进行调度优化，提高生产效率，降低成本。通过多目标优化，满足各车间的加工需求，使系统达到整体最优，并具有更强的柔性。

第一章，绪论。对国内外并行机作业车间调度问题、序列依赖准备时间调度问题以及多资源约束车间调度问题研究现状进行系统分析，做出总结，在文献分析的基础上建立研究框架，确定研究内容和思路。

第二章，相关问题描述与理论基础。将 SDST 分类讨论，说明本文所讨论的同一机器不同工件之间 SDST-PMJSS 问题与同一工件在不同机器间发生转化的 SDST-PMJSS 问题特点。详述 NSGA2 与 ABC 算法，并给出多目标优化的四种评价指标。

第三章，基于混合 NSGA2 的多目标 SDST-PMJSS 问题研究。根据第二章的分类情况选择最小化最大总完工时间、最小化总延迟时间、最小化总准备时间作为优化目标，对第一种 SDST-PMJSS 问题建立数学模型以更清晰分析问题的结构和特点。根据问题特点提出混合 NSGA2，并通过相关算例来验证混合算法的有效性。

第四章，多资源约束多目标 SDST-PMJSS 问题研究。在第二章所提的第二种 SDST-PMJSS 问题的基础上增加工序加工需要多种辅助资源这一限制条件，建立相关数学模型，并提出带有两阶段解码策略的混合 NSGA2，通过算例验证算法的有效性。

最后，研究总结与展望。对全文进行总结以及对未来工作提出展望。

全文技术路线图如图 1-1 所示。

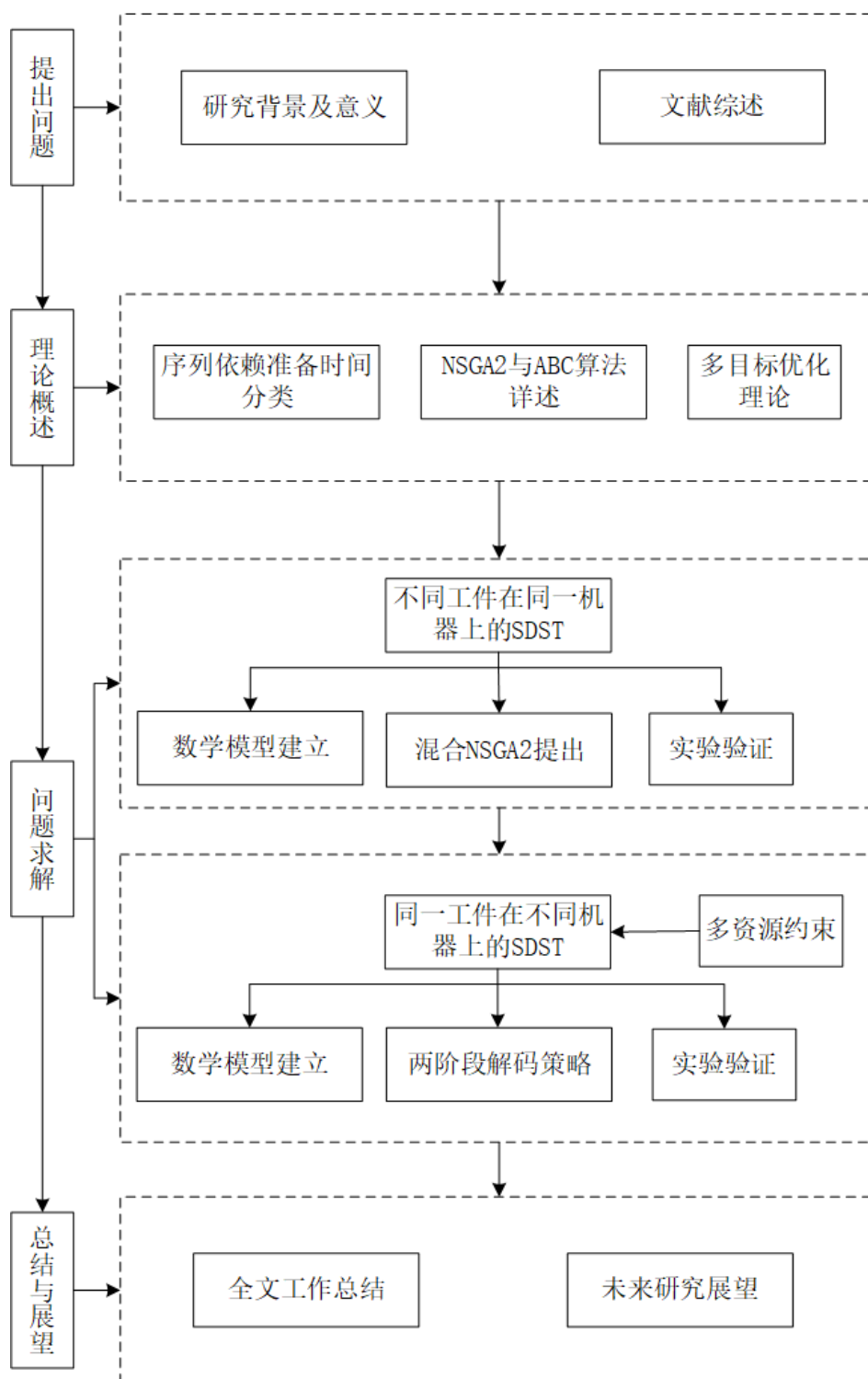


图 1-1 技术路线图

1.5 主要的创新点

在深入分析并行机作业车间调度问题后，根据在实际运用中出现的约束条件，选择最小化最大总完工时间、最小化总延迟时间、最小化总准备时间作为优化目

标,在经典并行机作业车间调度模型的基础上建立考虑序列依赖准备时间多目标并行机作业车间调度问题数学模型,对问题特点进行详细分析。

(1) 对序列依赖准备时间分类,选择不同工件在同一机器和相同工件在不同机器上的两种序列依赖准备时间进行研究。利用 NSGA2 和 ABC 控制参数少、操作简单以及融合度高并且在多目标优化中已广泛应用等特点,提出带有人工蜂群搜索策略的混合 NSGA2,使其能有效求解所提调度优化问题。

(2) 在第二种序列依赖准备时间的多目标并行机作业车间调度问题的基础上增加多资源约束这一条件,提出带有两阶段解码策略的混合 NSGA2,求解过程中不仅要考虑工件加工顺序,还要对辅助资源合理分配以满足加工要求。

第二章 相关问题描述与理论基础

2.1 引言

本章对全文讨论的序列依赖准备时间进行描述,介绍了所需要的相关理论基础。首先,对准备时间按三种不同的方式进行分类。然后,详细描述多并行机作业车间环境下带序列依赖准备时间的调度问题。接着,对本文需用到的 NSGA2 和 ABC 算法进行描述,对其特点进行总结归纳。最后,描述多目标优化中的基本理论并给出多目标优化算法的评价指标。

2.2 考虑准备时间的车间调度问题分类

近年来,准备时间对调度问题影响的重要性受到学术界和工业领域的广发关注,对此已有诸多研究。[Allahverdi 和 Soroush^{\[75\]}](#)指出准备时间是准备必要资源时所需要的时间,如机器开始加工任务前的准备时间。在许多现实情境中,设置通常发生在工件转移的过程中。同样,设置也包括设置机器夹具或固定装置;在夹具或固定装置里放置工件进行定位;对待加工工件所需的工具进行获取、调整、归还原位;清理和检查加工系统中的材料。

2.2.1 按照准备时间与序列相关性分类

在文献中,将带有准备时间的调度问题分为两类:准备时间与序列有依赖关系和准备时间与没有依赖关系^[76],而准备时间又可以通过批处理和非批处理两种加工方式产生,具体分类如图 2-1 所示。

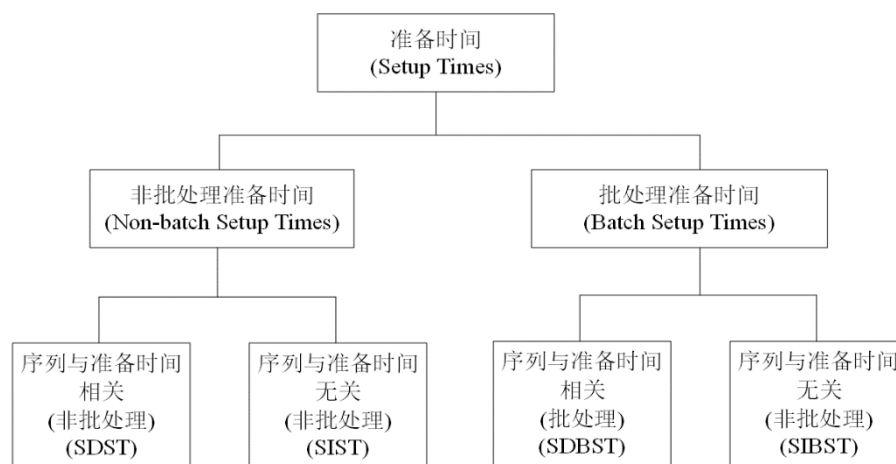


图 2-1 准备时间分类

(1) 序列与准备时间相关（非批处理）（Sequence Dependent Setup Times, SDST）。即加工不同种工件时机器需要的准备时间，此时序列的准备时间与正要加工工序和紧前工序同时相关。如纺织染色工厂，染料器具的清洁程度取决于先前染色的工序和即将染色的工序有关。

(2) 序列与准备时间无关（非批处理）（Sequence Independent Non-batch Setup Times, SIST）。此时序列的准备时间仅取决于当前任务，与紧前工序无关。如，简单的机加工车间，加工工艺具有一定程度的通用性。

(3) 序列与准备时间相关（批处理）（Sequence Dependent Batch Setup Times, SDBST）。

(4) 序列与准备时间无关（批处理）（Sequence Independent Batch Setup Times, SIBST）。如在半导体行业，通常将工件进行批处理，在将作业类型分组成批分配时会产生批量准备时间。

准备时间不产生利润但是又真实存在并且具有一定影响的，在如今高速发展的工业环境中起着非常重要的作用。企业在减少准备时间的同时需要增加盈利能力、核心竞争力、顾客满意度等。因此，对序列相关准备时间这一问题讨论是非常有意义的。

2.2.2 按照考虑准备时间调度问题目标分类

作业车间调度是一个多目标决策问题，Mizuno 和 Deeratanasrikul^[77]在对带有准备时间的作业车间调度问题综述研究中将调度目标进行归纳，具体内容如下所示：

C_{max}	最大完成时间
E_{max}	最大提前交付时间
T_{max}	最大延迟交付时间
\bar{C}	平均流经时间
$\sum(w_j) \cdot U_j$	加权总延迟工件
$\sum S_{ij}$	总准备时间
W_T	机器总负荷
W_M	机器最大负荷
Π	最小化总成本
Π'	最小化准备时间和库存成本和
$\sum(\omega') \cdot E_j + \sum(\omega'') \cdot T_j$	加权总提前时间与加权总延迟时间
$\sum \omega' \cdot C_{max} + \sum \omega'' \cdot T_j$	加权完成时间与加权总延迟时间和

$$\sum \omega_j \cdot C_j + \sum S_{ij} \quad \text{加权总完工时间与总安装时间和}$$

从整体上来看，目标函数基本分为单目标和多目标；又可以分为基于时间因素的目标，如交货期，完工时间，安装时间等；基于成本因素的目标如库存成本和提高机器利用率的如机器总负荷和机器最大负荷等。这些目标有的相互制约、相互矛盾，有的又有很强的相关性，然而目前实际生产中，调度的目标已经不再满足于单一的传统的目标，而是着眼于更能增加经济效益、降低成本，同时又能获得最大满意度的目标。因此，多目标调度问题的研究具有一定的实际意义，并且要根据具体情况来确定每个目标在总目标中的权重和优先级来选择最优解。

2.2.3 按照工序排序和机器转换的准备时间分类

(1) 不同工件之间的准备时间

在此类别下所考虑的准备时间体现在，按照工艺要求每台机器加工不同工件时，每个工件和其相邻工件之间产生一个准备时间用于调整，包括清理机器、更换夹具等，该准备时间是序列相关的，下面通过一组实例表 2-1 进行说明：

表 2-1 不同工件之间准备时间实例

机器	M_1			M_2			M_3		
工件	J_1	J_2	J_3	J_1	J_2	J_3	J_1	J_2	J_3
J_1	0	2	1	0	3	4	0	3	2
J_2	3	0	2	1	0	4	3	0	3
J_3	3	1	0	4	3	0	1	2	0

如上表所示在机器 M_1 上加工的三个工件：假如 J_1 是 J_2 的紧前工件， J_2 开始加工前的准备时间为 2；假如 J_2 是 J_1 的紧前工件， J_1 开始加工前的准备时间为 3；如果一个工件的若干工序安排在同一机器上加工，并且互相属于紧前或紧后工序，则机器在此加工过程中不需要调整，准备时间为 0。由于准备时间不产生任何收益同时会影响加工效率，因此调度过程中要使总准备时间最小化来提高整体效率，降低加工过程中不产生经济效益的时间。

(2) 工件在机器间转换产生的准备时间

在此类别下所考虑的准备时间体现在工件在机器之间转换产生的准备时间，设置活动包括机器所需资源的组装、软件下载、刀具校准和工件移动等，因此还

需要序列依赖的准备时间来拆卸原始机器以及组装和校准新机器以用于工件的传入^[78]以下通过一组甘特图来说明：

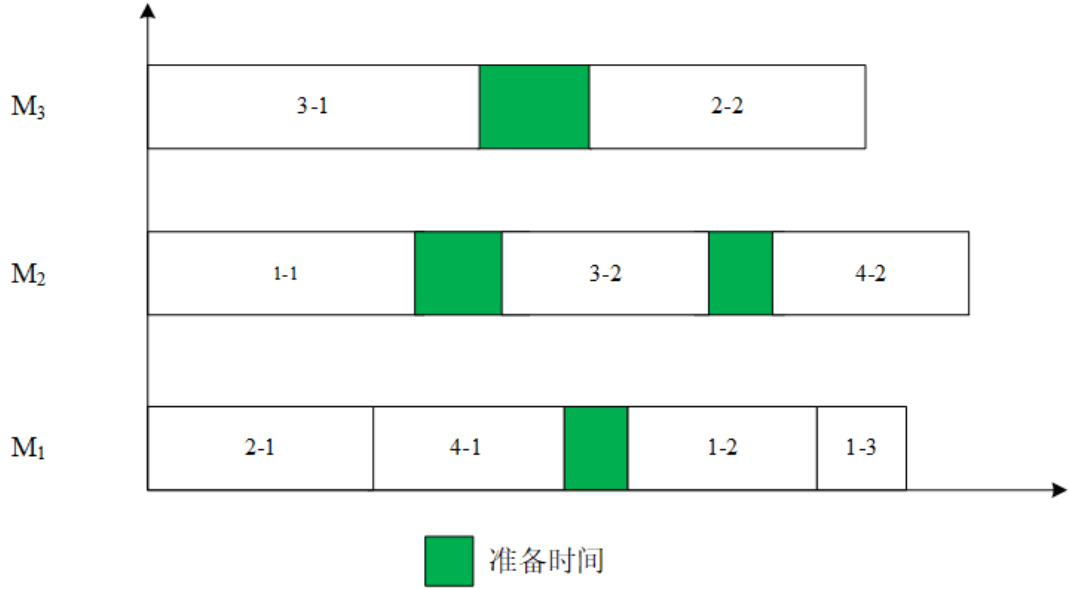


图 2-2 3 台机器 3 个工件调度方案示例甘特图

在此调度方案中，机器选择和工序排序为 $(O_{21}, O_{41}, O_{12}, O_{13}, M_1)$ ， $(O_{11}, O_{32}, O_{42}, M_2)$ ， (O_{31}, O_{22}, M_3) 。工件 J_1 的加工机器顺序为 $M_2 \rightarrow M_1 \rightarrow M_1$ ，发生机器转换，在加工工序 O_{12} 前产生一个准备时间；工件 J_2 的加工机器顺序为 $M_1 \rightarrow M_3$ ，此时产生一个准备时间；工件 J_3 的加工机器顺序为 $M_3 \rightarrow M_2$ ，发生机器转换，产生一个准备时间，且准备活动可以在工序 O_{31} 加工过程中进行；工件 J_4 的加工机器顺序为 $M_1 \rightarrow M_2$ ，发生机器转换，产生一个准备时间。工序 O_{21} 和工序 O_{41} 分别属于工件 J_2 和工件 J_4 的第一个工序，因此加工两个工序间不产生准备时间。

如上所述，学者们对序列依赖准备时间的研究已有多分类，每种类别可对应不同的实际问题，拥有不同的调度背景，本文对第三种分类的序列依赖准备时间进行讨论，问题中的准备时间提前已知并且确定。

2.3 SDST-PMJSS 问题描述

PMJSS 调度问题可定义如下：在调度系统中，有 k 台机器 $M = \{M_1, M_2, \dots, M_k\}$ ； i 个工件的集合 $J = \{J_1, J_2, \dots, J_i\}$ ；每一个工件有 j 个工序，工序集合 $O = \{O_{i1}, O_{i2}, \dots, O_{ij}\}$ ， O_{ij} 表示工件 i 的第 j 个工序。机器集合 M 中，能加工同一工序的机器作为一个机器集 $M_k \in M$ ，每一个机器集中的机器数 $m \geq 1$ 。

若 $m > 1$ ，即机器系统中出现并行机，所以每一个在并行机器阶段加工的工序 O_{ij} 都有一系列加工机器 $M_{ij} \subseteq M_k \subseteq M$ 可供选择，并且 $M_{ij} \neq \emptyset$ 。

并行机作业车间调度问题中，各工件的工序只能由相应的一台或多台机器加工，各工件在不同机器上的加工时间不同，且机器在加工工件时根据工艺顺序会产生不同的准备时间。因此，该问题包括两个子问题，一是要求确定各个机器上工件工序的分配（Machine Selection, MS）；二是各个机器上工序的加工顺序（Operations Sequencing, OS）同时满足优化目标要求。如此，制造系统的柔性会增强，求解难度增大。如表 2-2 中给出的并行机作业车间各工序加工时间示例，工序 O_{11} 可选的加工机器为 (M_2, M_3) ，对应的加工时间分别为 $(4, 6)$ ；工序 O_{23} 可选的加工机器为 (M_4, M_5, M_6) ，对应的加工时间分别为 $(7, 10, 13)$ 。

表 2-2 并行机作业车间各工序加工时间示例

		M_A		M_B		M_B	
		M_1	M_2	M_3	M_4	M_5	M_6
J_1	O_{11}	-	4	6	-	-	-
	O_{12}	-	-	-	4	3	5
	O_{13}	6	-	-	-	-	-
J_2	O_{21}	5	-	-	-	-	-
	O_{22}	-	7	8	-	-	-
	O_{23}	-	-	-	7	10	13
J_3	O_{31}	-	-	-	3	4	6
	O_{32}	-	4	5	-	-	-
	O_{33}	-	-	-	7	9	8

SDST-PMJSS 问题是在 PMJSS 问题的基础上增加 SDST 的条件，本文所讨论的 SDST 具体描述见节 2.2.3。在确定了各工序分配的机器和各机器上工序的排序后，需要根据 SDST 按照工序顺序确定各工序的序列依赖开始准备时间和工序开始加工时间，从而得到各调度目标值。

2.4 快速非支配排序遗传算法（NSGA2）

上世纪九十年代 Deb 和 Srinivas 提出了非支配排序遗传算法(Non-dominated Sorting Genetic Algorithms, NSGA)，这是一种基于 Pareto 最优概念的遗传算法。

在解决实际问题中由于 NSGA 算法存在计算复杂度高、缺少精英策略以及需要人为指定共享半径 Q_{share} 这些缺点, Deb^[59]于 2002 年在 NSGA 算法基础上提出了快速非支配排序遗传算法 (NSGA2), 该算法相比 NSGA 算法做了以下改进:

(1) 改进了快速非支配排序, 使算法复杂度由原先 $O(mN^3)$ 降为 $O(mN^2)$, 在种群规模变大时效果尤为明显。

(2) 不再采用人为指定共享半径 Q_{share} 机制, 使用拥挤度比较算子加以代替。计算每个个体的拥挤距离, 其值和快速非支配排序共同决定可行解集中哪个解更优, 使同级较优个体均匀分配, 保持了种群的多样性。

(3) 引入精英策略, 在交叉变异产生子代后, 将父代与子代合并成一个新的种群, 其规模是父代的两倍。通过精英策略再筛选出新的父代, 如此扩大了解的采样空间, 有利于精英个体保留至后代中, 增加种群的多样性。

2.4.1 快速非支配排序

该算法是对规模为 n 的种群进行分层, 种群集合为 N , 对所有个体进行快速非支配排序, 使每个个体获得第一个属性: 非支配序 i_{rank} (个体所处层级数), 具体步骤如下:

Step1: 所有个体表示为 x_i , 设 $i=1$ 。

Step2: 对于所有的 $x_{i'}, i'=1, 2, \dots, n, i' \neq i$, 比较个体 x_i 与个体 $x_{i'}$ 之间的非支配关系。

Step3: 如果所有个体 $x_{i'}$ 均被个体 x_i 支配, 将个体 x_i 标记为非支配个体, 并存于第一支配层 Z_1 。

Step4: 令 $i=i+1$, 转 Step2, 找到所有非支配个体, 并存于 Z_1 。

Step5: 将 Z_1 中所有个体从 N 中剔除, 转 Step1, 直至种群所有个体被分层, 产生层级集合 $Z = \{Z_1, Z_2, \dots, Z_n\}$ 。

2.4.2 拥挤距离

为了改进 NSGA 算法中需要人为指定共享半径 Q_{share} 这一问题, NSGA2 采用了拥挤距离加以改进: 将拥有目标极值的个体, 即落在边界的个体, 设为无穷大距离, 对中间的个体根据每一个目标函数计算该个体两侧作为顶点所组成的长方形周长, 定义为拥挤距离, 全部拥挤距离通过每一个目标的距离值累加和得到。在确定所有个体的拥挤距离后, 每个个体获得第二个属性, 拥挤度 i_d 。个体的拥挤度越大, 其周围密度越稀疏, 个体数量越少, 该个体越占优。

2.4.3 拥挤度比较算子

拥挤度比较算子可以在个体被快速非支配排序后在同级中筛选出更优个体，保证了种群的多样性。经过非支配排序和拥挤度的确定每个个体都有两个属性：非支配序 i_{rank} 和拥挤度 i_d 。通过这两个属性可以定义拥挤度算子：在两个个体 x_i 与 $x_{i'}$ 比较过程中若其中一个个体满足以下任意条件，则该个体胜出，并进入下一操作。

(1) 个体 x_i 的非支配序小于个体 $x_{i'}$ 的非支配序，即 $i_{rank} < i'_{rank}$ 。

(2) 若个体 x_i 与个体 $x_{i'}$ 非支配序相同，而个体 x_i 拥挤距离更大，即 $i_{rank} = i'_{rank}$ ， $i_d > i'_d$ 。

2.4.4 精英策略

NSGA2 算法程序将规模为 N 的父代种群 P_t 与交叉变异产生的子代种群 Q_t 直接合并成规模为 $2N$ 的种群 R_t 。此时的种群 R_t 包含了 P_t 与 Q_t 所有的个体，通过对所有个体的快速非支配排序产生非支配集合 Z_n ，计算每个个体的拥挤度距离，此时所有个体均获得非支配序和拥挤度两个属性。将各非支配集中个体按照所在非支配集顺序 $\{Z_1, Z_2, \dots, Z_i\}$ 依次放入新的父代种群 P_{t+1} 中，若 P_{t+1} 中个体数小于 N ，则继续将下一非支配集个体放入 P_{t+1} ，重复同样操作直至 P_{t+1} 中个体数大于等于 N 。如果在未将 Z_i 中所有个体添加进 P_{t+1} 中， P_{t+1} 中的个体数已大于等于 N ，则通过拥挤度比较算子选择 Z_i 中部分个体，使 P_{t+1} 中的个体数等于 N 。

2.4.5 NSGA2 的特点

NSGA2 自从提出来至今，在多目标优化邻域取得了成功，成为目前最流行的多目标优化算法之一。由于 NSGA2 是一种带有精英策略的快速非支配排序遗传算法，因此 NSGA2 保留了遗传算法适应性强、编码灵活等优点。选择、交叉、变异三个算子都是随机进行操作，使算法使用随机方法进行搜索最优解。在 NSGA2 算法中，新子代种群 Q_{t+1} 在通过遗传算子产生后直接进入下一次迭代。因此传统 NSGA2 受遗传算子自身局限性影响，在交叉变异过程中算法不能保证对个体进行邻域搜索，导致算法局部搜索能力不足，解的分布不均匀，容易陷入局部最优，在针对车间调度等离散型多目标优化问题时比较明显。

NSGA2 流程如图 2-3 所示。

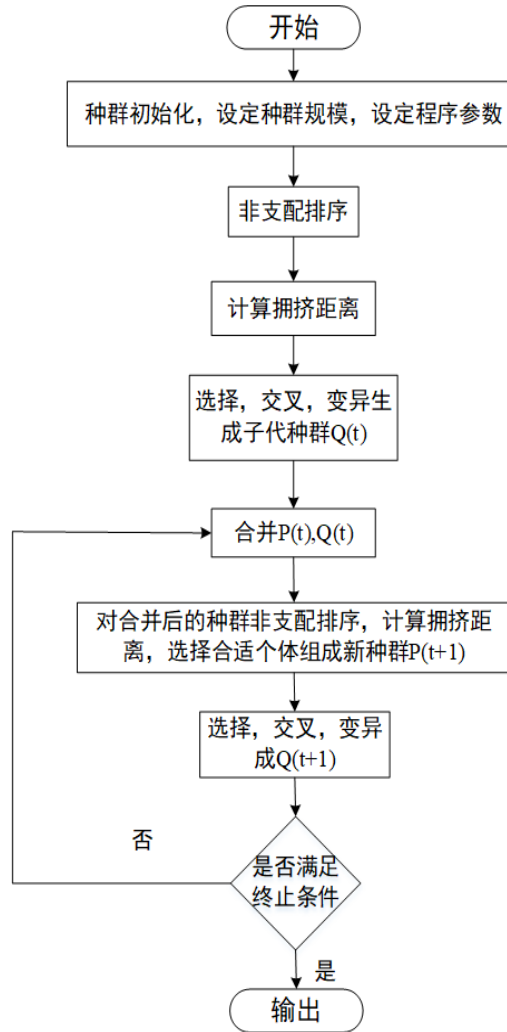


图 2-3 NSGA2 算法流程

2.5 人工蜂群算法（ABC）

Karaboga^[65]于 2005 年系统地提出人工蜂群算法（Artificial Bee Colony Algorithm, ABC），在和其他智能优化算法性能比较后指出 ABC 算法适合处理工程中的高维问题。由于 ABC 算法在很多方面体现出优良的性能，已经成为仿生智能邻域的一种重要的优化算法。

2.5.1 ABC 算法相关参数定义及流程

基本人工蜂群算法适用于求解单目标的连续优化问题，算法过程模拟了蜜蜂采蜜，主要由蜜源、采蜜蜂、跟随蜂和侦查蜂组成。其中蜜源的位置对应优化问题的可行解，蜜源的蜜量对应可行解的质量，用适应度函数值大小表示。每一个蜜源对应一个采蜜蜂，采蜜蜂在旧蜜源的周围随机发现新蜜源，通过蜜量选择新

旧蜜源，并将蜜源信息传递给跟随蜂。计算选择采蜜蜂传来蜜源的概率，跟随蜂通过轮盘赌的方式选择，并在所选择的蜜源附近进行邻域搜索，从旧蜜源位置产生新的蜜源位置。当采蜜蜂在一个蜜源周围搜索 $limit$ 次，仍然没有产生新蜜源，该蜜源位置被舍弃。采蜜蜂转化为侦查蜂，按照随机方式寻找新的蜜源。

根据文献^[79]算法中相关参数可定义如下：

蜜蜂总数为 N ，采蜜蜂的规模 N_c ，跟随蜂的规模为 N_g 通常认为 $N_c = N_g$ 。每个蜜源即可行解 $X_i (i \in [1, 2, \dots, N])$ ，解的维度为 D 即优化问题的参量个数。可行解适应度函数值 $fFitness(i)$ ，跟随蜂选择蜜源概率 P_i ，最大迭代次数 $maxCycle$ ，第 n 代采蜜蜂种群 $X(n)$ ，搜索限定次数 $limit$ ，搜索次数 Bas 。

相关公式如下：

$$X_{ij} = L_{ij} + \Omega * (U_{ij} - L_{ij}) \quad (2-1)$$

式(2-1)表示初始可行解的产生方式， Ω 为 $[0, 1]$ 之间的一个随机数， j 取值于 $\{1, 2, \dots, D\}$ 。 X_{ij} 表示第 i 个蜜源的位置， U_{ij} 与 L_{ij} 分别表示蜂群在蜜源 X_i 位置搜索的上界和下界。

$$X_{vj} = X_{ij} + \omega * (X_{ij} - X_{kj}) \quad (2-2)$$

式(2-2)表示跟随蜂在蜜源 X_i 位置周围继续搜索产生新蜜源位置的方式，其中 ω 为 $[-1, 1]$ 之间均匀分布的随机数， $k \in [1, 2, \dots, N]$ ， k 随机生成，且 $k \neq i$ 。

$$P(i) = \frac{fFitness(i)}{\sum_{i=1}^N fFitness(i)} \quad (2-3)$$

采蜜蜂将蜜源 X_i 信息传递给跟随蜂后，跟随蜂按式(2-3)计算选择该蜜源的概率 P_i 。

$$fFitness(i) = \begin{cases} \frac{1}{f(X_i)}, & f(X_i) > 0 \\ 1 + abs(f(X_i)), & f(X_i) \leq 0 \end{cases} \quad (2-4)$$

式(2-4)表示可行解适应度计算方式，其中 $f(X_i)$ 表示目标函数值，当目标函数值为负数时，取其绝对值。原则上拥有较高适应度函数值的蜜源会被选择，适应度函数值对蜜源是否保留起决定性作用。

$$X_i(n) = L_{ij} + \Omega * (U_{ij} - L_{ij}) \quad (2-5)$$

跟随蜂结束搜索后，如果搜索次数 $Bas(i) \geq limit$ ，该蜜源 X_i 仍未被更新，该蜜源所对应的采蜜蜂变为侦查蜂，同时舍弃当前蜜源，侦查蜂按式(2-5)随机产生一个新蜜源来代替该蜜源。其中， Ω 为 $[0, 1]$ 之间的一个随机数。

$$X_v = \begin{cases} X_i, & fFitness(i) \geq fFitness(v) \\ X_v, & fFitness(i) < fFitness(v) \end{cases} \quad (2-6)$$

式(2-6)为三种蜜蜂的贪婪选择方式,新旧蜜源中适应度函数值大的蜜源将被保留。

人工蜂群算法中,每次搜索过程中有着以下三种不同的步骤:

- (1) 采蜜蜂到达蜜源位置并评估蜜源的丰富程度。
- (2) 跟随蜂在接收到采蜜蜂传来的蜜源信息后,选择蜜源。
- (3) 侦查蜂产生,寻找新的蜜源,以代替旧蜜源。

三种蜜蜂在选择蜜源时包含如下过程:

- (1) 跟随蜂使用全局选择来发现蜜源最丰富的区域,该区域存在最优解。
- (2) 采蜜蜂和跟随蜂使用局部选择,并根据局部信息从旧蜜源位置邻域搜索产生新蜜源位置。
- (3) 三种蜜蜂均使用贪婪选择,当每个蜜蜂发现新产生的蜜源比旧蜜源丰富,则舍弃旧蜜源,否则保留旧蜜源。
- (4) 侦查蜂使用随机选择,当蜜源被舍弃后侦查蜂随机产生新蜜源。

将人工蜂群算法用于解决优化问题时,蜜蜂采蜜行为与优化问题的对应关系如表 2-3 所示:

表 2-3 蜜蜂采蜜行为与优化问题对应关系

蜜蜂采蜜行为	优化问题
蜜源位置	优化问题的可行解
蜜源的丰富程度	可行解的质量(适应度函数值)
寻找蜜源速度	可行解优化速度
最丰富的蜜源	优化问题最优解

人工蜂群算法就是通过以上描述的基本组成元素,模仿蜜蜂采蜜行为,三种蜜蜂所得信息相互交流,每种蜜蜂进行邻域搜索以实现局部寻优,通过贪婪选择,在迭代过程中实现全局寻优。因此,人工蜂群算法是一种借助启发式的搜索策略,不仅具有局部寻优能力,也具有全局寻优能力的现代启发式智能搜索算法。

2.5.2 算法流程

根据上述,人工蜂群算法步骤如下:

Step1: 初始化种群,设定各个控制参数,蜜蜂 N (采蜜蜂和跟随蜂的数量设为 $N/2$),最大迭代次数 $maxCycle$,迭代次数 $iter=0$,最大搜索限定次数 $limit$,搜索次数 $Bas=0$ 。

Step2: 开始所有蜜蜂均为侦查蜂,按照随机方式产生 N 个初始可行解。

Step3: 计算每个可行解的适应度函数值, 将蜜蜂分为采蜜蜂和跟随蜂两种, 适应度函数值在前 50% 的可行解对应的蜜蜂为采蜜蜂, 后 50% 的为跟随蜂。

Step4: 采蜜蜂进行在初始可行解周围邻域搜索, 按式(2-1)产生新解, 并计算其适应度函数值。

Step5: 采蜜蜂按式(2-6)对新老蜜源进行贪婪选择, 若选择新蜜源被选择, 更新当前采蜜蜂所在位置, $Bas(i)=0$, 否则 $Bas(i)=Bas(i)+1$ 。

Step6: 按式(2-3)计算跟随蜂选择采蜜蜂所反馈蜜源的概率。

Step7: 跟随蜂根据所选择的蜜源按式(2-2)产生新蜜源, 同时转化为采蜜蜂, 对所选择的蜜源邻域搜索, 并计算其适应度函数值。

Step8: 跟随蜂按式(2-6)贪婪选择 Step7 产生的新老蜜源, 如果新蜜源适应度函数值较高, 则新蜜源代替旧蜜源, 新蜜源对应的跟随蜂变为采蜜蜂更新 $Bas(i)$ 。

Step9: 判断 $Bas(i)$ 是否大于最大搜索限定次数 $limit$: 如果小于, 转至 Step10; 否则, 当前蜜源还没有被更新, 放弃当前蜜源, 两种蜜蜂变为侦查蜂。侦查蜂按式(2-5)产生新蜜源替换被放弃蜜源。

Step10: 记录当前搜索蜜蜂找到的最优解。

Step11: 判断是否满足迭代次数, 若满足, 算法结束, 输出最优值; 否则, 转至 Step3。

人工蜂群算法流程图如图 2-4 所示:

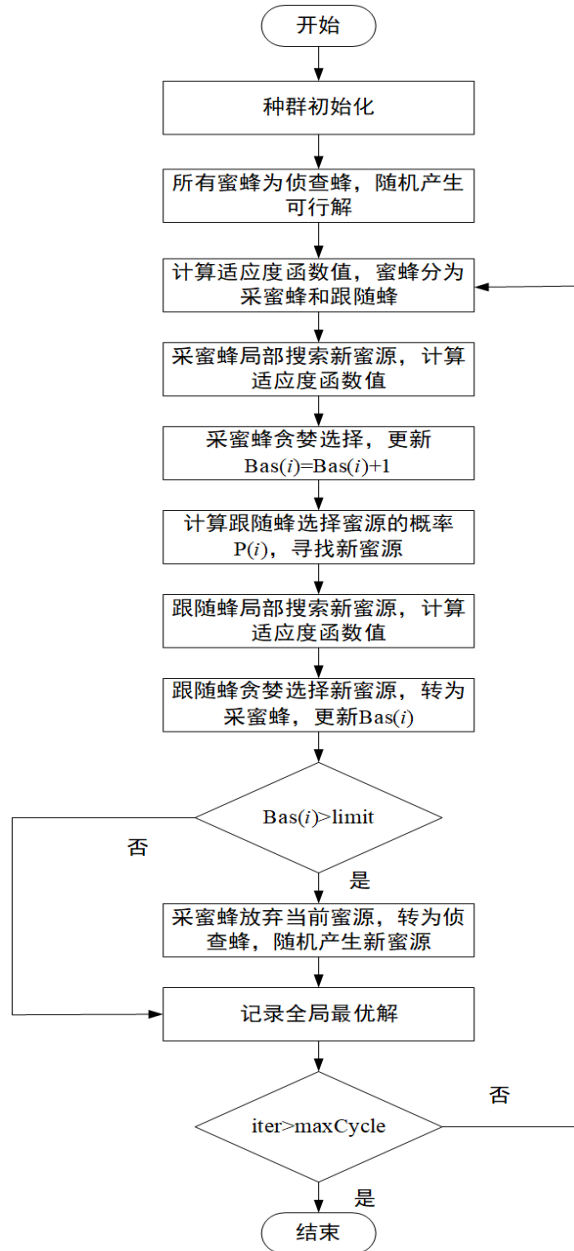


图 2-4 人工蜂群算法流程图

2.5.3 ABC 算法的特点

人工蜂群算法自提出以来受到了广泛关注,已成功应用于工程应用、神经网络、图像识别等多个领域,根据文献^{[80][81]}人工蜂群算法主要特点体现在以下几个方面:

(1) 算法是通过长期对自然界蜜蜂群体智能行为观察得到启发,并应用于解决优化问题,同人工智能技术相结合产生的。

(2) 人工蜂群算法是一种广义的邻域搜索算法,每次全局搜索过程都有三种蜜蜂进行局部搜索,算法三个主要部分信息能够相互交流。

(3) 算法搜索能力较好, 侦查蜂舍弃旧解, 随机找到新解这一特征减弱了算法对种群规模的依赖性, 保证了种群多样性。

(4) 算法参数较少, 参数可设置范围广, 减小了实验人员因参数设置所产生的偏差。随机值和固定值均可作为算法初值的设定, 算法适应性较强。

(5) 算法原理简单, 容易实现, 通用性较强, 可以很好地与实际问题相结合, 有效解决复杂优化问题。

2.6 多目标优化

2.6.1 多目标优化问题基本理论

十九世纪九十年代末, 法国经济学家 Pareto 提出多目标优化问题 (Multi-objective Optimization Problem, MOP), 该问题是一类具有多个目标冲突的问题, 重难点在于如何处理多个目标, 目标之间可能相互制约, 相互影响。目前在解决多目标问题时, 加权法和 Pareto 支配法是最常使用方法。加权法是通过给各个目标附以权重, 通过线性方法将多目标优化问题转为单目标优化问题加以求解。Pareto 支配法是将寻找到的不同的解决方案根据 Pareto 支配概念存入 Pareto 档案集, 在这些方案中至少有一个目标值要比其他目标值更优。

不失一般性, 多目标优化问题具有如下定义:

定义 1 多目标优化问题

$$\begin{aligned} \text{Max / Min } F(x) &= (f_1(x), f_2(x), \dots, f_k(x)) \\ \text{s.t. } g_i(x) &\leq 0, i = 1, 2, \dots, m \\ h_j(x) &= 0, j = 1, 2, \dots, n \\ x &\in X \end{aligned} \quad (2-7)$$

式(2-7)描述了多目标优化问题的数学表达, 其中 $x = (x_1, x_2, \dots, x_n)^T$ 是决策空间 X 中的一个 n 维决策变量, $F(x)$ 是 k 维目标向量, $(f_1(x), f_2(x), \dots, f_k(x))^T$ 是 k 个相互矛盾的目标向量, $g_i(x)$ 为 m 个不等式约束条件和 $h_j(x)$ 为 n 个等式约束条件, 决定了 x 的取值范围。多目标优化的目的是为了求出 $x^* = (x_1^*, x_2^*, \dots, x_n^*)^T$ 满足所有约束条件使得目标函数 $F(x^*)$ 取得最优值。

定义 2 可行解

对于决策空间 X 中某个解 x , 满足式(2-7)中约束条件 $g_i(x) \leq 0, (i = 1, 2, \dots, m)$ 和 $h_j(x) = 0, (j = 1, 2, \dots, n)$, 则称 x 为一个可行解。

定义 3 可行解集合

决策空间 X 中所有可行解组成的集合, 定义为 X_f , $X_f \subseteq X$ 。在多目标优化问题中, 具有多个需要同时优化的子目标, 并且各子目标之间可能会相互影响,

各子目标的量化标准也不同,在优化了一个目标性能后,其他目标性能会有所损失,因此不存在一个可行解使所有目标同时达到最优。

定义 4 Pareto 支配 (Pareto Dominate)

设 X_a , X_b 为两个决策向量, $X_a, X_b \in X_f$, 在最小化问题中, 若满足以下条件:

$$\begin{cases} f_k(X_a) \leq f_k(X_b), \forall k \in \{1, 2, \dots, n\} \\ f_l(X_a) < f_l(X_b), \exists l \in \{1, 2, \dots, n\} \end{cases} \quad (2-8)$$

根据式(2-8)可知, X_a 的所有子目标值 $f_k(X_a)$ 都不比 X_b 的子目标值 $f_k(X_b)$ 差, 并且 X_a 中至少存在一个目标函数值小于 X_b , 则 X_a 优于 X_b , 又称为 X_a 支配 X_b , 记作 $X_a \prec X_b$ 。

定义 5 Pareto 最优解 (Pareto Optimal Solution)

若一个解 b 满足

$$\nexists a \in X_f : a \prec b \quad (2-9)$$

即在当前可行解集 X_f 中不存在任何一个可行解 a 能够支配可行解 b , 则称 b 为 Pareto 最优解。Pareto 最优解又称为非劣解或非支配解, 且 Pareto 最优解不唯一, 多个 Pareto 最优解构成 Pareto 最优解集 (Pareto Optimal Solution Set)。

定义 6 Pareto 最优解集 (Pareto Optimal Solution Set)

所有 Pareto 最优解构成的集合称为 Pareto 最优解集 C^* , 满足:

$$C^* = \{x \in X_f \mid \nexists a \in X_f : a \prec x\} \quad (2-10)$$

由式(2-10)可知, 集合 C^* 中的解均是相互非支配的, 两两之间不存在支配关系。

定义 7 Pareto 前沿 (Pareto Front)

Pareto 最优解集中每个最优解通过对应的目标函数映射构成了该问题 Pareto 前沿, 或 Pareto 前沿面。

在存在多个 Pareto 最优解的情况下, 如果问题的相关信息没有足够多, 很难对最优解进行取舍, 因此对于多目标优化问题需要找到尽可能多的 Pareto 最优解使其更均匀地接近多目标问题实际的 Pareto 前沿。

2.6.2 多目标优化评价指标

衡量多目标优化算法的性能, 主要从算法收敛速度、得到解的质量、相对于其他算法的有效性等来进行。通过文献查阅, 本文对多目标优化算法性能的评价从以下三个指标来进行:

(1) 平均理想距离 MID^[82] (Mean Ideal Distance), 该指标反映 Pareto 解到理想点的距离, 用于衡量算法的收敛速度。MID 值越低, 表示算法收敛速度越快, 性能越好。计算公式如式(2-11)所示。

$$MID = \frac{\sum_{i=1}^n c_i}{n} \quad (2-11)$$

其中 n 表示非支配解集数, $c_i = \sqrt{f_{1i}^2 + f_{2i}^2 + f_{3i}^2}$, (f_{1i}, f_{2i}, f_{3i}) 表示第 i 个非支配解集的三个目标函数值。

(2) 非支配解的传播 SNS^[83] (Spread of Non-dominance Solution), 该指标反映算法产生非支配解的多样性。SNS 值越高, 算法生成解的多样性越好。计算公式如式(2-12)所示。

$$SNS = \sqrt{\frac{\sum_{i=1}^n (MID - c_i)^2}{n-1}} \quad (2-12)$$

(3) 支配率 POD^[84] (Percentage of Domination), 该指标反应算法支配其他算法的能力。POD 值越大, 该算法的支配其他算法能力越强, 解的质量更好, 即算法越有效, 该指标作为主要评价指标。计算公式如式(2-13)所示。

$$POD(X', X'', \dots, X^n) = \frac{|\{x_{kj} \in X_i\} \mid \exists x_{kj} \in X_j, i \in j: x_{ki} \leq x_{kj}\}|}{|X_j|} \times 100\% \quad (2-13)$$

POD 是在 CM (Coverage metric) 的基础上提出来的, CM 表示一个 Pareto 解集 X' 支配另一个 X'' 的程度。如果相互比较算法超过两个, 所有解集都要被分析。

2.7 本章小结

本章描述全文所讨论的准备时间的特点, 详细介绍了相关理论知识。首先, 对准备时间进行分类, 说明本文所讨论的不同工件在同一机器上工件切换时和同一工件在不同机器上发生机器转换时的两种序列依赖准备时间。然后, 详细介绍了 NSGA2 和 ABC 算法中重要定义以及算法流程。最后, 介绍了多目标优化的问题的基本理论, 给出了包含三个主要指标 MID、SNS、POD 的多目标优化算法的评价指标概念以及计算方式。

第三章 基于混合 NSGA2 的多目标 SDST-PMJSS 问题研究

3.1 引言

本章研究了不同工件在同一机器上加工的序列相关准备时间问题。首先，建立以最小化总完工时间、最小化总延迟时间、最小化总准备时间为目标的 SDST-PMJSS 的调度模型。然后，将 ABC 算法修改为 MOABC 算法，并且将其中三个搜索部分引入 NSGA2，设计出带有人工蜂群搜索策略的混合 NSGA2。最后，通过算例实验将 NSGA2 与 NSGA2 和 MOABC 进行效果对比。

3.2 多目标 SDST-PMJSS 问题模型构建

3.2.1 问题假设

与传统的 JSS 问题相比较，SDST-PMJSS 问题是并行机车间调度问题与作业车间调度问题的混合，并在此基础上考虑了序列依赖准备时间。在本章中考虑的是不同工件在同一机器上的序列依赖准备时间，同一个加工机器加工诸多不同种类工件，此加工环境适用于多品种小批量的生产模式。该问题需满足以下基本约束条件：

- (1) 工序的加工顺序必须按照工件加工的工艺要求。
- (2) 同一时刻一个工序只能由一台机器加工，且一台机器同一时间只能加工一个工件，加工开始之后不允许中断。
- (3) 不同工件之间没有优先级关系。
- (4) 每个工件在相应机器上的加工时间已知，但是在并行机器中哪台机器上加工未知，在零时刻所有机器都可以加工。
- (5) 工件的加工工艺顺序已知。

3.2.2 相关变量和参数定义

- | | |
|-----|----------------------------------|
| i | 工件索引号, $i = 1, 2, 3, \dots, n$ |
| j | 工序索引号, $j = 1, 2, 3, \dots, n_i$ |
| k | 机器索引号, $k = 1, 2, 3, \dots, m$ |

n_i	工件 i 的工序数
K_{ij}	工件 i 的第 j 个工序可选机器集合, $ K_{ij} \geq 1$
N_k	机器 k 上的工序总数
λ	在机器 k 上加工的工序顺序编号
O_{ijk}	工件 i 的第 j 个工序在机器 k 上加工
JT_{ijk}	工件 i 的第 j 个工序在机器 k 上开始加工时间
JS_{ijk}	工件 i 的第 j 个工序在机器 k 上序列依赖准备开始时间
P_{ijk}	工件 i 的第 j 个工序在机器 k 上的加工时间
T'_{ik}	工件 i' 先于工件 i 在机器 k 上开始加工前工件 i 准备时间
$S_{(i'j')(ij)k}$	工序 $O_{i'j'}$ 先于工序 O_{ij} 在机器 k 上开始加工, 工序 O_{ij} 开始加工前的准备时间
C_{ij}	工序 O_{ij} 的完工时间
C_i	工件 i 的完工时间
C_{max}	最大完工时间
D_i	工件 i 的交货期
d_i	工件 i 延迟交货时间
$\alpha_{(i'j')(ij)k}$	如果工序 O_{ij} 和工序 $O_{i'j'}$ 都被分配到机器 k 上, 工序 O_{ij} 紧随工序 $O_{i'j'}$ 加工时取值为 1, 否则为 0
$\beta_{ijk\lambda}$	如果工序 O_{ij} 被分配到机器 k 上第 λ 个位置取值为 1, 否则为 0
φ	一个极大的正整数

3.2.3 数学模型构建

通过第二章节 2.2 和节 2.3 对 SDST-PMJSS 问题的分析, 结合实际生产调度情况, 选择最大完工时间最小、总延迟时间最小和总准备时间最小作为三个目标函数。最大完工时间反映了企业的生产效率; 在实际工业活动中, 订单准时交付以满足客户需求显得尤为重要, 拖期交付会给企业带来相应的惩罚, 因此总延迟时间最小可以给企业带来更小的经济或者信誉损失; 准备时间在生产过程中不产

生经济价值,但是又不能忽略,因此降低生产过程中的准备时间可以给企业减少生产成本。以下为多目标 SDST-PMJSS 问题数学模型:

目标函数:

$$f_1 = \min C_{\max} \quad (3-1)$$

$$f_2 = \min \sum_{i=1}^n d_i \quad (3-2)$$

$$f_3 = \min \sum_{i=1}^n \sum_{j=1}^{n_i} \sum_{k=1}^m \alpha_{(i'j')(ij)k} \cdot S_{(i'j')(ij)k} \quad (3-3)$$

约束条件:

$$d_i \geq D_i - C_i \quad \forall i \quad (3-4)$$

$$JT_{ijk} + P_{ijk} + S_{ijk} \leq C_{ijk} \quad \forall i, j, k \quad (3-5)$$

$$C_{ij} \leq JT_{i(j+1)} \quad \forall i, j \quad (3-6)$$

$$C_{in_i} \leq C_{\max} \quad \forall i \quad (3-7)$$

$$S_{(i'j')(ij)k} = T_{i'ik} \quad i \neq i', j \neq j' \quad \forall i, i', j, j', k \quad (3-8)$$

$$JT_{ijk} - S_{ijk} = JS_{ijk} \quad (3-9)$$

$$\begin{aligned} & \left(JT_{i'j'k} + P_{i'j'k} \right) \cdot \beta_{i'j'k\lambda} + S_{(i'j')(ij)k} \cdot \alpha_{(i'j')(ij)k} \leq JT_{ijk} \cdot \beta_{ijk\lambda+1} + \varphi \left(\alpha_{(i'j')(ij)k} - 1 \right) \\ & i \neq i', j \neq j' \quad \forall i, i', j, j', \lambda, k \end{aligned} \quad (3-10)$$

$$\begin{aligned} & \left(JT_{ijk} + P_{ijk} \right) \cdot \beta_{ijk\lambda} + S_{(ij)(i'j')k} \cdot \alpha_{(ij)(i'j')k} \leq JT_{i'j'k} \cdot \beta_{i'j'k\lambda+1} + \varphi \left(\alpha_{(ij)(i'j')k} - 1 \right) \\ & i \neq i', j \neq j' \quad \forall i, i', j, j', \lambda, k \end{aligned} \quad (3-11)$$

$$\alpha_{(ij)(i'j')k} + \alpha_{(i'j')(ij)k} \leq 1 \quad i \neq i', j \neq j' \quad \forall i, i', j, j', k \quad (3-12)$$

$$\sum_{\lambda=1}^{N_k} \sum_{k=1}^m \beta_{ijk\lambda} = 1 \quad \forall i, j, \lambda, k \quad (3-13)$$

$$JT_{ijk}, S_{ijk}, P_{ijk}, d_i \geq 0 \quad \forall i, j, k \quad (3-14)$$

其中,式(3-1)、式(3-2)和式(3-3)描述的是本文研究的三个最小化目标函数,分别为最小化总完工时间,最小化总延迟时间和最小化总准备时间。式(3-4)定义了每个工件延迟时间的计算方法。式(3-5)表示工序 O_{ij} 在机器 k 上开始加工前的

准备时间、开始加工时间与加工时间的和不大于其完工时间。式(3-6)表示工件的工艺顺序,同一工件的上一个工序加工完成后下一个工序才能开始加工。式(3-7)表示各工件工序的完工时间不大于加工系统总完工时间。式(3-8)表示各工序的序列依赖准备时间计算方法。式(3-9)表示每个工序在序列依赖准备时间结束后立即开始加工。式(3-10)和式(3-11)表示同一机器上相邻两个不同工件工序的加工顺序,即在同一机器上,上一个工件工序加工完成,且机器加工下一工件工序的准备时间结束后,下一工件工序才能开始加工。若调换两个工序在机器上的加工顺序,则两工件工序之间的序列依赖准备时间不同。式(3-12)加强说明同一机器上同一个位置只能分配一个工序。式(3-13)表示同一个工序在同一时间只能分配到一台机器上的一个加工位置加工。式(3-14)各工序的开始时间、加工时间、序列依赖准备时间以及工件的延迟时间均为正数。

3.3 混合 NSGA2 求解多目标 SDST-PMJSS 问题

根据第二章节 2.4 与节 2.5 对 NSGA2 和 ABC 算法的描述可知,NSGA2 通过交叉、变异的方式从父代种群产生子代种群后直接与父代种群合并,再通过非支配排序和拥挤度比较选出和父代种群相同规模的种群。与遗传算法相比,NSGA2 改进了其搜索策略,降低了算法复杂度,增大了采样空间^[85]。由于 NSGA2 得到的最优解来自于合并后的种群,所以最优解的质量受子代种群质量的影响,并且该算法存在着局部搜索能力不足的缺点。人工蜂群算法的三个主要部分其实是在限定范围内的一种邻域搜索过程,通过一定的寻优策略引导算法向更好的解搜索。如果在 NSGA2 交叉变异产生子代种群后将人工蜂群算法引入,用子代种群引导蜂群再一次寻优,通过贪婪选择可以得到更好的解。

3.3.1 混合原理

NSGA2 具有很强的全局搜索能力,但是种群主要通过遗传算子从父代中产生子代种群实现更新,因此在解决实际工程问题时,容易陷入局部最优而无法收敛到全局最优解。如果将遗传算子得到的子代种群进一步优化,以一定的策略引导其向更好解的方向进行搜索,即可保证在算法收敛的基础上改善这一问题,提高解的质量。将 ABC 算法应用到多目标优化问题上时,需要对其引入 Pareto 支配的概念和拥挤度算子改进成多目标人工蜂群 (Multi-objective Artificial Bee Colony, MOABC) 算法,由于只是更改了解的适应度函数值计算方法以及优劣解的评估标准,因此保留了 ABC 算法的三部分相互交流、适应性强以及有效进行局部搜索等特点。为了充分利用两种算法的优点,在 NSGA2 算法交叉变异后将

MOABC 中的采蜜蜂、跟随蜂和侦查蜂引入，对子代种群再次引导性搜索，设计出混合 NSGA2，在保证种群多样性的同时，使其尽可能找到多的优质解。

混合 NSGA2 核心原理为：将 NSGA2 作为主算法，发挥 NSGA2 的全局搜索能力。在主程序设定种群规模 N 、交叉和变异参数，NSGA2 最大迭代次数 $maxCycle$ 、搜索限定次数 $limit$ 。NSGA2 种群初始化后先行生成父代种群 P_t ，通过交叉、变异操作生成子代种群 Q_t 。MOABC 将不再作为整体算法进行搜索，而是采蜜蜂、跟随蜂和侦查蜂按照既定的步骤与寻优方式在子代种群 Q_t 的基础上生成新 Q_t 。NSGA2 合并 P_t 与新 Q_t 生成 R_t ，经过精英策略选择出新 P_{t+1} ，进行非支配排序选出最优解。如此反复，在混合 NSGA2 运行结束后输出全局最优解。

3.3.2 种群初始化

在进化算法中，种群初始化是非常重要的一个步骤，目的是产生初始解，为后续算法提供搜索方向。目前采用较多的是随机初始化方式，这种方式操作简单，但是可能产生质量较低的初始解，而初始解的质量又会影响算法的收敛速度和运行时间。

本文采用混合初始化策略，使用随机初始化规则和最小加工时间规则，两种规则所占比例为 1:1，即 50% 的种群个体采用随机初始化规则产生，50% 的种群个体采用最小加工时间规则产生。这种混合策略旨在保证初始种群多样性的同时提高个体质量。

3.3.3 个体的编码

多目标 SDST-PMJSS 是一个复杂的离散型组合优化问题，而 ABC 算法是处理连续函数的最优化方法。因此需要对工件调度进行适当编码，以便后续 ABC 算法的处理。良好的编码方式可以提高算法的执行效率、减少不可行解的产生和提高计算精度。使用进化算法求解 SDST-PMJSS 问题，调度方案通过染色体的形式展现，染色体的可行性、有效性以及问题空间对应解空间的完整性是编码的关键。该问题调度方案应该包括每个机器上工序的分配以及各机器上工序的加工顺序。因此，本章采用 MSOS 分段式编码方案，将机器选择和工序排序分开编码，一个完整的染色体编码包含 MS 和 OS 两部分。在一个调度问题中，染色体的总长度 $T = 2T_0$ ， T_0 为工序总数，在一个完整染色体中前 $T/2$ 的位置为 MS 部分，后面的部分为 OS 部分。

在 MS 部分采用基于机器的编码方式，每个基因代表机器的序号。以表 2-2（见节 2.3）为例， O_{12} 对应的加工机器为 (M_4, M_5, M_6) ，在 MS 部分 O_{12} 的基因是 5，表示的是可选机器集中第二台机器 M_5 。这种编码方式保证了后续产生新

种群操作后染色体依然是可行解。MS 排序中编码示例如图 3-1 所示。

M ₅			M ₄			M ₆		
2	5	1	1	3	4	4	3	6
O ₁₁	O ₁₂	O ₁₃	O ₂₁	O ₂₂	O ₂₃	O ₃₁	O ₃₂	O ₃₃

图 3-1 机器选择部分编码示例

在 OS 部分采用基于工序的编码方式，这种编码方式柔性高，适用于不同规模大小的工件工序，对不同环境问题的适应性强。编码中每个基因数字代表工件的工件号，工件号出现的次数代表该工件工序的加工顺序。如图 3-2 所示工序排序部分编码示例中，第一个数字 1 表示工件 J_1 的第一个工序 O_{11} ，第二个数字 1 表示 J_1 的第二个工序 O_{12} 。

J ₁	J ₃	J ₂	J ₃	J ₂	J ₁	J ₂	J ₃	J ₁
1	3	2	3	2	1	2	3	1
O ₁₁	O ₃₁	O ₂₁	O ₃₂	O ₂₂	O ₁₂	O ₂₃	O ₃₃	O ₁₃

图 3-2 工序排序部分编码示例

一个完整的可行调度编码方案是将上述 MS 部分和 OS 部分连接起来，据上述，每个部分的长度均为 9 对应整个加工系统的 9 个工序。因此，该实例完整的编码方案长度为 18。在 MS 部分工序 ($O_{11}, O_{12}, O_{13}, O_{21}, O_{22}, O_{23}, O_{31}, O_{32}, O_{33}$) 将被分配到机器 ($M_2, M_5, M_1, M_1, M_3, M_4, M_4, M_3, M_6$) 上加工；在 OS 部分所有工序的加工顺序为 ($O_{11}, O_{31}, O_{21}, O_{32}, O_{22}, O_{12}, O_{23}, O_{33}, O_{13}$)。

该分段编码实例表示的调度方案为：(O_{11}, M_2) (O_{31}, M_4) (O_{21}, M_1) (O_{32}, M_3) (O_{22}, M_3) (O_{12}, M_5) (O_{23}, M_4) (O_{33}, M_6) (O_{13}, M_1)。

3.3.4 解码

编码是由两部分组成，而根据节 3.2 建立的问题模型，解码不仅要考虑工序和机器，还需要在工件开始加工前确定序列依赖准备时间，相关步骤如下：

Step1: 从左向右依次读取基因，获得各个工序对应的加工机器。

Step2: 通过加工机器矩阵 JM , 加工时间矩阵 PT , 各机器上所有工件之间的准备时间矩阵 ST , 各个工件的交付时间。

Step3: 由 Step1 和 Step2 获得了每个工序的加工信息, 对每个机器上的加工工序按照工序约束进行排序。

Step4: 由 Step3 获取的机器上工序排序信息, 获得每个工件加工前的准备时间。根据工序编号和在机器上加工位置编号判断工序是否为机器上的第一道工序, 或者同一机器上工序与其紧前工序是否属于同一工件, 满足其中任意条件则工序在机器上的序列依赖准备时间等于 0; 如果两个条件均不满足, 根据当前工序的工序号判断其所属的工件编号以及在机器上的加工位置, 计算该工序在工件上的紧前工序完工时间 t_1 和在同一机器上紧前工序完工时间 t_2 , $\Delta t = t_1 - t_2$ 。若 $\Delta t < 0$, 该工序的开始准备时间等于 t_2 , 则该工序的开始加工时间等于 $t_2 + S_{ijk}$; 若 $\Delta t = 0$, 该工序的开始准备时间等于 t_1 或 t_2 , 则该工序的开始加工时间等于 t_1 或 t_2 任取一值加上 S_{ijk} ; 若 $\Delta t > 0$, 该工序的开始准备时间等于 $t_1 - S_{ijk}$, 则该工序的开始加工时间等于 t_1 。

Step5: 检查是否每个基因均被读取, 如果是, 则结束; 否, 跳转 Step1 解码。

3.3.5 交叉操作

在自然界, 交叉操作是遗传过程中非常重要的一个步骤。在交叉过程中, 两个父代染色体从某一基因位置切断, 切断后的部分进行组合形成两个新子代染色体。交叉操作可以将父代优良基因遗传到子代, 能够扩大解的搜索空间, 提高搜索速度。本文采用分段交叉的方式分步进行, 在机器选择部分采用 RPX 交叉方式, 在工序排序部分采用 IPOX 交叉方式。这两种交叉操作分开进行, 能使父代优良个体保留下来, 同时确保子代个体都是可行解, 下面将介绍两种交叉操作。

RPX 交叉操作过程如图 3-3 所示, 在交叉过程中保留工序排序部分顺序不变。其中, 其中 $P1$, $P2$ 代表两个父代染色体的机器部分, $C1$ 和 $C2$ 代表两个子代染色体的机器部分。具体步骤如下所示:

Step1: 在区间 (0,1) 之间随机生成与 $P1$, $P2$ 长度一致的数组 T , 数组中每一个值对应一个位置上的基因。

Step2: 根据公式(3-15)计算值 pf , 记录 $P1$ 中大于 pf 的基因位置, 将 $P1$ 中该位置上的基因复制到 $C1$, 保持位置不变。记录 $P2$ 中大于 pf 的基因位置, 将 $P2$ 中该位置上的基因复制到 $C2$, 保持位置不变。

Step3: 将 $P1$ 中剩余基因复制到 $C2$, 将 $P2$ 中剩余基因复制到 $C1$ 。

$$pf = pf_{max} - \frac{pf_{max} - pf_{min}}{max\ Cycle} \times Cycle \quad (3-15)$$

式(3-15)中, pf_{max} , pf_{min} 分别表示数组 T 中的最大值与最小值, $Cycle$ 为当前迭代次数, $maxCycle$ 为总迭代次数。

T	0.81	0.90	0.12	0.63	0.16	0.27	0.54	0.39	0.42
P1	2	5	1	1	3	4	4	3	6
P2	3	6	1	1	2	4	4	5	6
C1	2	5	1	1	2	4	4	5	6
C2	3	6	1	1	3	4	4	3	6

图 3-3 RPX 交叉过程

IPOX 交叉操作过程如图 3-4 所示, 在交叉过程中保留机器选择部分顺序不变。其中 $P1$, $P2$ 代表两个父代染色体的工序部分, $C1$ 和 $C2$ 代表两个子代染色体的工序部分, $JobSet(1)=\{1\}$, $JobSet(2)=\{2,3\}$ 。具体步骤如下所示:

Step1: 将所有工件随机分成两个非空工件集合 $JobSet(1)$ 和 $JobSet(2)$ 。

Step2: 将 $P1$ 中所有包含 $JobSet(1)$ 的基因复制到 $C1$, 将 $P2$ 中所有包含 $JobSet(2)$ 的基因复制到 $C2$, 两次复制过程均保证基因位置不变。

Step3: 将 $P2$ 中所有包含 $JobSet(2)$ 的基因复制到 $C1$, 将 $P1$ 中所有包含 $JobSet(1)$ 的基因复制到 $C2$, 两次复制过程均保证基因位置不变。

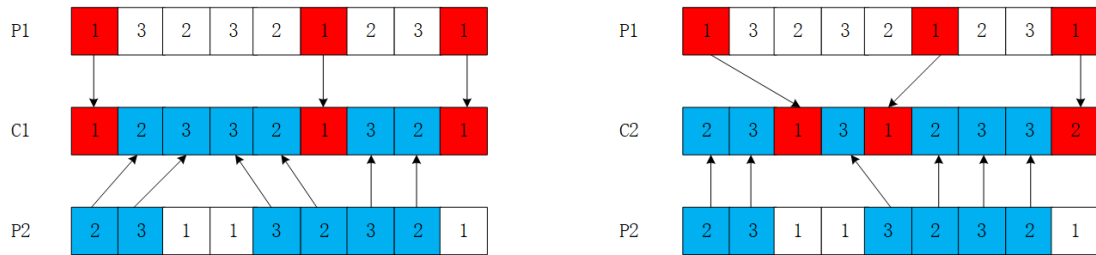


图 3-4 IPOX 交叉过程

3.3.6 变异操作

在交叉操作完成后需要进行变异, 目的是增强算法的局部搜索能力, 避免过早地陷入局部最优, 有利于维持种群的多样性。根据分段编码的特点, 在机器选择部分的每个基因数字均代表给各工序安排的加工机器编号。由于各工序可选的加工机器数大于等于 1, 所以机器选择部分的基因数字可以用工序可选机器集中其他机器编号更换, 具体实例如图 3-5 所示。在进行工序排序部分变异操作时,

保留机器选择部分不变，随机选择工序排序部分两个基因，相互交换位置。这种操作虽然随机改变了基因位置，但是相应的工艺顺序也发生改变，因此可以保证生成的子代都是可行解，具体操作实例如图 3-6 所示。

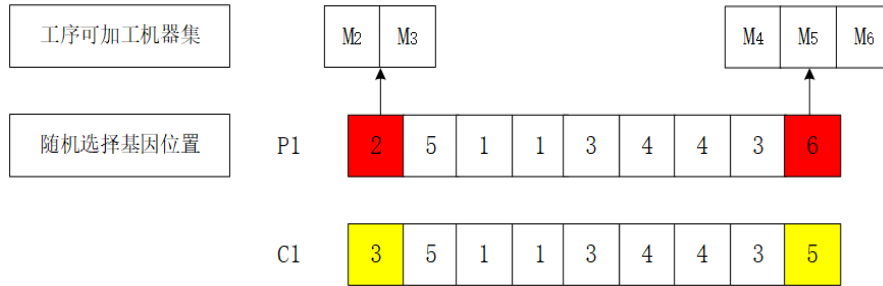


图 3-5 机器选择部分变异操作

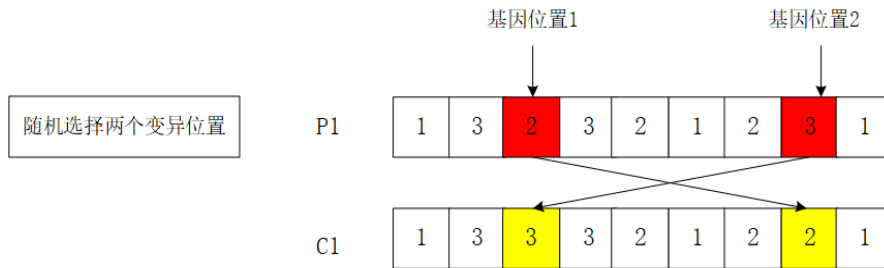


图 3-6 工序排序部分变异操作

3.3.7 外部存档的更新

在多目标优化算法中，外部存档是一个非常重要的部分，通常用来存储和维护算法流程中产生的 Pareto 最优解。外部存档的容量通常要提前设置，维护过程具有如下步骤：

(1) 通过 Pareto 支配原则判断每一轮产生的 Pareto 最优解与外部存档中解的支配关系，新解被支配，则舍弃新解，不进入外部存档。

(2) 若新解与外部存档内的解无支配关系，则新解进入外部存档中，更新外部存档中解的数量 $C(i) = C(i) + 1$ 。

(3) 若新解支配旧解，则将新解加入代替其所支配的旧解。

(4) 判断外部存档中解的数量是否大于外部存档容量，若大于则计算外部存档中所有解的拥挤距离，将拥挤距离最小的解剔除，以保证外部存档中解的数量不超过其容量上限。

3.3.8 多目标 ABC 算法

基本 ABC 算法适用于求解连续性的单目标优化问题，将 ABC 算法应用到多目标优化问题时需要将基本 ABC 算法修改成多目标 ABC 算法^[86]，下面给出修改策略。

与其他智能优化算法类似，单目标 ABC 算法计算每个解 X_i 的适应度函数值，以此作为优劣解的评估标准。在多目标优化问题中，将 Pareto 支配的概念引入来计算每个解的适应度值，如果 $X_2 \prec X_1$ ，则认为 X_1 优于 X_2 ， X_1 在种群中的可支配解数量 $DS(i)$ 加 1。解的适应度函数值按式(3-16)计算。

$$fFitness(i) = \frac{DS(i)}{N} \quad (3-16)$$

在多目标优化问题中，不能通过简单的适应度函数值来判断解的优劣。因此，蜂群通过 Pareto 支配准则来评估邻域搜索后新旧解的优劣关系。若新解支配旧解，则新解更优， $Bas(i)$ 值加 1。

根据 Pareto 支配准则找出种群中所有非劣解，非劣解按照式(3-17)产生邻域解，种群中其他的可行解按照式(2-1)产生邻域解，并按照贪婪准则选出新解。

$$X_{vj}^* = X_{vj} + \Omega * (X_{vj} - X_{lj}) \quad (3-17)$$

通过以上更改策略使多目标 ABC 算法在寻优过程中，将种群中的非劣解作为全局信息指引蜂群的搜索方向，其他的解再随机生成新解以进行贪婪选择，如此不仅可以保证种群的多样性，还能够避免所有蜜蜂过于集中，使所有解的分布更加均匀。

3.3.9 混合 NSGA2 算法步骤

根据上述对算法关键部分的描述，混合 NSGA2 具有以下步骤：

Step1: 初始化种群，设定种群规模 N ，设定交叉率 P_c ，变异率 P_m ，NSGA2 最大迭代次数 $maxCycle$ ，外部存档的容量 C ，搜索限定次数 $limit$ 。初始化外部存档集。

Step2: 种群个体非支配排序，获得个体非支配序 i_{rank} 。计算个体拥挤度距离，获得个体拥挤度值 i_d 。生成初始父代 P_t 。

Step3: 对 P_t 中的个体进行交叉操作，工序排序部分 IPOX 交叉，机器选择部分 RPX 交叉。交叉完成后对个体运用混合变异策略完成变异操作，生成子代 Q_t 。

Step4: 根据子代 Q_t 规模蜜蜂分为采蜜蜂与跟随蜂。

Step6: 在子代 Q_t 中根据 Pareto 支配原则，找出非劣解，加入外部存档。

Step7: 采蜜蜂根据式(3-17)对所有非劣解进行邻域搜索。剩余的解由采蜜蜂

根据式(2-1)对其进行邻域搜索。计算每个新解适应度函数值,采蜜蜂通过 Pareto 支配原则进行贪婪选择新旧解,更新 $Bas(i)$ 。

Step8: 根据式(2-3)计算跟随蜂选择采蜜蜂传递来的解的概率 $P(i)$ 。

Step9: 跟随蜂通过轮盘赌的方式选择解,并根据式(2-2)寻找新解,转化为采蜜蜂进行邻域搜索,通过 Pareto 支配原则进行贪婪选择,更新 $Bas(i)$ 。

Step10: 判断 $Bas(i)$ 是否大于最大搜索限定次数 $limit$: 如果小于,保留该解;否则,放弃该解,采蜜蜂变为侦查蜂。侦查蜂按照式(2-5)产生新解替换被放弃解。

Step11: 蜂群搜索部分结束产生新子代 Q_t 。

Step12: 合并父代 P_t 与新子代 Q_t , 生成 R_t , 种群规模为 $2N$ 。

Step13: 计算 R_t 中非支配序和个体拥挤距离。根据精英策略删除多余个体,生成 P_{t+1} 。通过 Pareto 支配关系选出 P_{t+1} 中 Pareto 最优解,更新外部存档。

Step14: 判断全局搜索是否满足迭代次数 $maxCycle$, 若满足,输出外部存档;否则,转至 Step3。

混合 NSGA2 流程如图 3-7 所示。

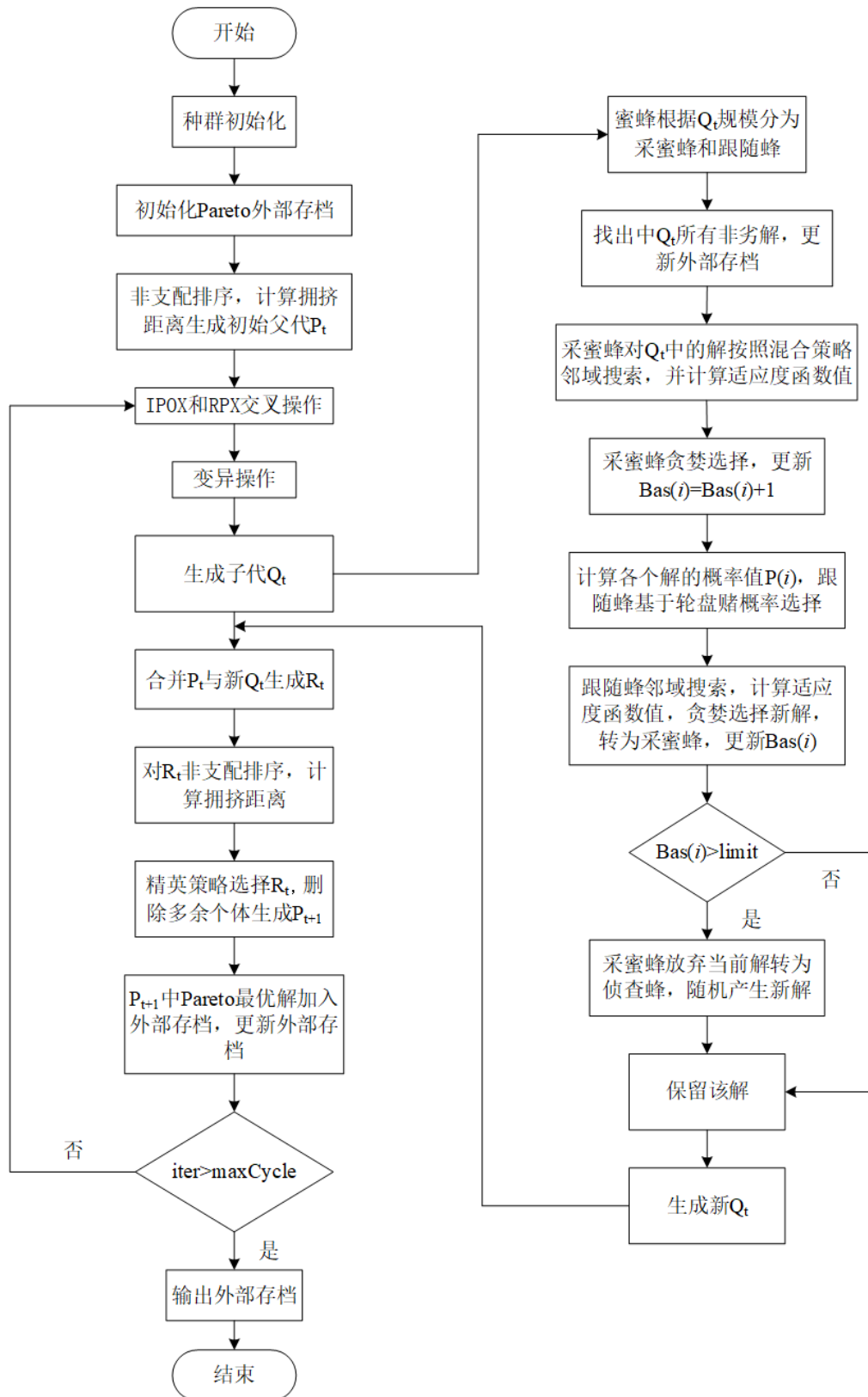


图 3-7 混合 NSGA2 流程图

3.4 实验设计与结果分析

3.4.1 仿真环境

上述混合 NSGA2 通过 MATLAB 语言编程实现，仿真软件为 MATLAB R2019b。运行环境为：AMD Ryzen 5 4600U CPU，主频为 2Ghz，内存为 16G，操作系统为 Windows 10。

3.4.2 实验数据设计

根据本文问题背景，通过查阅文献未发现已给出的不同工件之间的准备时间。因此，本文从 Fattahi instances 以及 Lawrence instances 中选出 20 组具有不同规模的算例作为工序和机器的实验数据基础，该组数据工件数在 3~20 之间，机器数在 3~10 之间，每个工件的工序数在 3~10 之间。如 10×10 工件组，共 10 个工件在 10 台机器上加工，每个工件有 10 道工序。根据相应数据的工件数与机器数，通过 MATLAB 生成 0~3 之间的随机整数作为序列依赖的准备时间。具体说明如表 3-1 中 5 台机器 4 个工件序列依赖准备时间示例所示。在 5 个工件，4 台机器的加工环境中，序列依赖准备时间表第一行表示机器编号，第一列表示工件编号，其中每一格中所有元素与工件总数相同。如第二行第二列中第一个 1 表示工件 1 和工件 2 若都在在机器 2 上加工，且工件 1 是工件 2 的紧前工件，则两工件之间的准备时间为 1。

表 3-1 5 台机器 4 个工件序列依赖准备时间示例

	1	2	3	4
1	[0,3,3,3,2]	[0,1,3,3,3]	[0,2,2,1,3]	[0,2,3,1,2]
2	[3,0,1,1,3]	[2,0,1,2,3]	[3,0,2,1,2]	[3,0,3,3,1]
3	[3,3,0,3,1]	[1,1,0,2,2]	[3,3,0,1,1]	[1,1,0,3,3]
4	[1,2,2,0,3]	[3,3,2,0,1]	[3,3,2,0,2]	[3,1,2,0,3]
5	[1,1,3,2,0]	[1,1,3,2,0]	[1,3,1,2,0]	[3,3,2,3,0]

3.4.3 参数设置

由于本文算法是用 NSGA2 控制全局搜索，监控蜂群对子代种群 Q_t 进一步优化，在蜂群阶段需要设置 $limit$ 。NSGA2 参数根据文献^[87]设置。MOABC 算法具有适应性强的特点，参数可设置范围广，因此算法参数设置对其性能影响较小。蜂群搜索部分中 $limit$ 对其影响相对较大，和种群规模相关，但只要不设置得过

小，就可以显著降低其对算法优化能力的影响^[88]，*limit* 具体设置参考文献^[89]。因此，混合 NSGA2 相关参数设置如表 3-2 所示。

表 3-2 混合 NSGA2 参数设置

种群规模	交叉率 P_c	变异率 P_m	$maxCycle$	外部存档容量	$limit$
50	0.9	0.1	100	30	10

3.4.4 实验结果与分析

本文将 NSGA2、MOABC 两种基本多目标优化算法作为对比算法，以验证混合策略解决序列依赖准备时间在同一机器不同工件之间的多目标 SDST-PMJSS 问题的有效性。为保证对比算法之间的公平性，其他两种算法与本文所提的混合 NSGA2 算法采用相同的编码与解码方案。每个实验例子的准备时间在各算法上相同，所有算法采用相同的交叉变异算子，MOABC 采用基本的搜索策略。每个实验例子在各算法上均独立运行 20 次，各算法参数统一设置。每个编号对应的数据得到 60 组 Pareto 解集，再通过 MID、SNS、POD 三种指标计算方式得出最终实验结果来对比各算法的性能，为了简化数据形式，表格中的 MID 值与 SNS 值均除以 10^5 ，相关评价指标详述见节 2.6.2。三种算法在 20 组算例上得到的多目标评价指标值如表 3-3 所示。

表 3-3 三种算法在 20 组算例上得到的多目标评价指标 (MID、SNS、POD) 值

$n \times m$		NSGA2			MOABC			混合 NSGA2		
		MID	SNS	POD	MID	SNS	POD	MID	SNS	POD
1	3×3	2.4355	2.4951	33.89	2.4355	2.4953	32.42	2.4354	2.4953	33.95
2	3×4	2.4355	2.4960	34.25	2.4355	2.4961	32.40	2.4355	2.4961	33.25
3	3×5	2.4354	2.4934	32.76	2.4354	2.4828	32.76	2.4354	2.4934	34.48
4	4×5	2.4354	2.4946	33.33	2.4354	2.4946	33.33	2.4354	2.4946	33.33
5	10×5	2.4350	2.4867	25.07	2.4349	2.4850	10.62	2.4351	2.4872	64.30
6	10×5	2.4349	2.4874	26.03	2.4349	2.4872	15.07	2.4350	2.4874	58.90
7	10×5	2.4348	2.4872	15.12	2.4348	2.4868	8.29	2.4350	2.4888	76.59
8	10×5	2.4350	2.4863	10.77	2.4349	2.4871	2.31	2.4348	2.4897	86.92
9	15×5	2.4348	2.4610	1.13	2.4346	2.4637	1.13	2.4347	2.4692	97.74
10	15×5	2.4346	2.4713	12.00	2.4345	2.4744	12.00	2.4347	2.4810	76.00
11	15×5	2.4349	2.4543	2.31	2.4348	2.4542	2.31	2.4349	2.4711	95.38
12	15×5	2.4345	2.4778	9.80	2.4342	2.4756	1.96	2.4348	2.4820	88.24
13	20×5	2.4343	2.4571	1.96	2.4343	2.4530	1.96	2.4345	2.4692	96.08
14	20×5	2.4343	2.4501	2.75	2.4342	2.4484	2.75	2.4344	2.4714	94.50
15	20×5	2.4342	2.4506	3.33	2.4343	2.4502	3.33	2.4344	2.4718	93.33
16	20×5	2.4341	2.4619	12.41	2.4343	2.4484	8.38	2.4345	2.4705	79.21
17	10×10	2.4343	2.4888	5.17	2.4340	2.4883	5.17	2.4340	2.4913	89.66
18	10×10	2.4343	2.4841	19.23	2.4346	2.4831	2.31	2.4341	2.4883	78.46
19	10×10	2.4341	2.4840	16.67	2.4346	2.4836	16.67	2.4342	2.4882	66.67
20	10×10	2.4341	2.4863	12.00	2.4344	2.4849	12.00	2.4343	2.4985	76.00

混合 NSGA2 在第 5 组 10×5 算例上求出最优解调度甘特图如图 3-8 所示, 其中绿色部分表示不同工件之间的准备时间。如图 3-8 中所示, 0 时刻所有机器均可以加工, 且加工的工序均为对应工件的第一个工序, 因此没有序列依赖准备时间。在机器 M_2 的前两个加工位置分别加工工件 J_4 的前两个工序 O_{41} 和工序 O_{42} , 则工序 O_{42} 开始加工前的序列依赖准备时间等于 0。机器 M_2 的第一个加工位置加工工件 J_9 的第二个工序 O_{92} , 此时机器 M_2 上加工的工件从 J_4 变为 J_9 , 因此在机器 M_2 上工序 O_{92} 的紧前工序是工序 O_{42} , 工序 O_{92} 的序列依赖准备时间 $S_{(4,2)(9,2)}=1$ 。由于 O_{91} 的完工时间等于 17, 而在机器 M_2 上, O_{42} 的完工时间等于 $P_{41} + P_{42} = 77 + 55 = 132 > 17$, 则工序 O_{92} 的序列依赖开始准备时间等于 132, 开始加工时间等于 133。

从表 3-3 三种算法在 3 个多目标评价指标（MID、SNS、POD）上的比较结果可以得出以下结论：

（1）在指标 MID 上，三种算法在每组实例上差值很小，表示算法的收敛速度差距不大。说明本文所提混合 NSGA2 没有对 NSGA2 和 MOABC 收敛速度造成破坏，并且将两个子算法收敛速度快的特点继承下来。

（2）在 4 个小规模实例上，混合 NSGA2、NSGA2、MOABC 三种算法在 SNS、POD 三个指标上表现差距不大。当实例规模变大，在编号 5、7、8、9 等 15 组实例上混合 NSGA2 的 SNS 值明显高于其他两种对比算法；16 组实例上混合 NSGA2 的 POD 值明显高于其他两种对比算法，尤其在编号 7、8、9、10 等 13 组实例上，混合 NSGA2 的 POD 值均超过了 70%。由此可见，混合 NSGA2 在解决多目标 SDST-PMJSS 问题上的能力比 NSGA2 和 MOABC 更优，混合算法的支配能力更强，能搜索到更多、质量更高的非支配解，在解的多样性上呈现了极大优势。这是因为混合 NSGA2 继承了 NSGA2 的全局搜索能力以及 MOABC 收敛速度快，搜索精度高等特点。使用采蜜蜂和跟随蜂对交叉变异产生的个体邻域搜索，进一步优化，侦查蜂产生新解替换被舍弃的劣解，突破 NSGA2 易陷入局部最优的问题。同时，在蜂群进一步优化阶段采用两种策略引导蜂群搜索，使解的分布更均匀，有效维护了解的多样性。

综上所述，在求解不同工件之间准备时间的多目标 SDST-PMJSS 问题时，本章提出的混合 NSGA2 在以 20 组标准算例为基础所产生的实例上测试，以 MID、SNS、POD 为评价指标进行结果对比，验证了本文所提出的混合 NSGA2 在求 SDST-PMJSS 问题时的有效性和优越性。

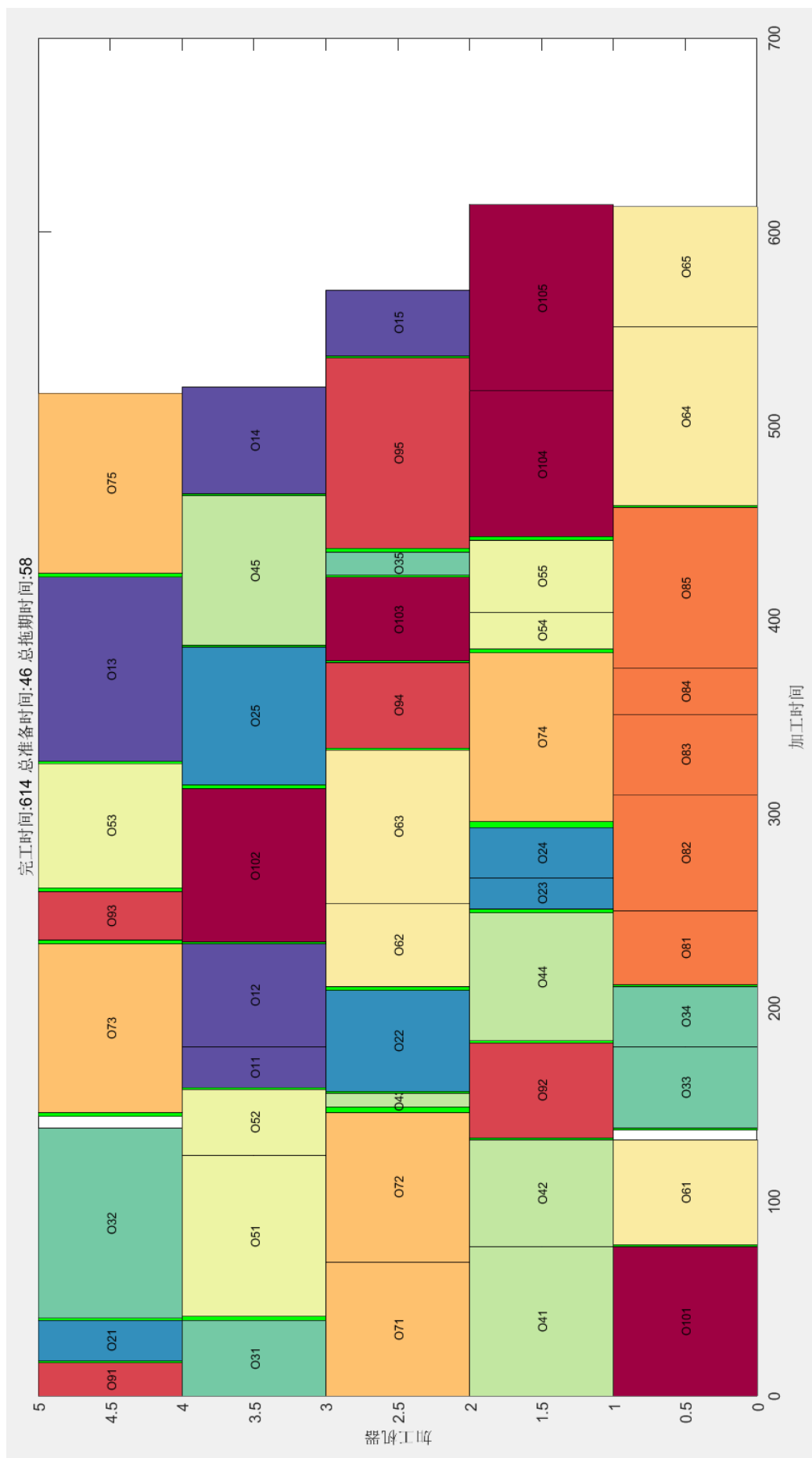


图 3-8 混合 NSGA2 在算例 10×5 的最优调度甘特图

3.5 本章小结

本章针对第二章所提出的不同工件之间在同一机器上的序列依赖准备时间进行讨论,提出混合 NSGA2 有效求解了多目标 SDST-PMJSS 问题。首先,建立了多目标 SDST-PMJSS 问题调度模型;然后根据第二章对 NSGA2 和 ABC 算法的优缺点提出混合 NSGA2。引入分段式编码策略、设计了针对该类型序列依赖准备时间的解码策略;在交叉、变异部分,采用一种新的 IPOX 和 RPX 混合交叉策略;引入 Pareto 支配和拥挤度比较算子,根据编码方式将 ABC 算法改为 MOABC 算法。将 MOABC 中采蜜蜂、跟随蜂、侦查蜂部分引入 NSGA2 中,对交叉、变异产生的子代中的个体邻域搜索,并贪婪选择,增强算法的搜索能力,提高解的多样性与质量;最后,通过仿真实验,以标准算例为基础生成不同工件之间在同一机器上的序列依赖准备时间的 SDST-PMJSS 问题进行测试,并与 NSGA2 和 MOABC 两种基本算法在 3 种多目标评价指标上进行对比,实验结果验证了混合 NSGA2 在求解该调度问题的有效性。

第四章 多资源约束多目标 SDST-PMJSS 问题研究

4.1 引言

本章在第三章的基础上研究了多资源约束多目标 SDST-PMJSS 问题的求解方法，SDST 发生于工件在机器间转换的过程中。首先，对多资源约束多目标 SDST-PMJSS 问题进行描述。然后，沿用第三章所选择的最小化总完工时间、最小化总延迟时间和最小化总准备时间作为调度目标，建立数学模型。接着，将第三章混合 NSGA2 进行修改，设计两阶段解码策略求解多资源约束多目标 SDST-PMJSS 问题。最后，设计相应的算例，将本章混合 NSGA2 与 NSGA2、MOABC 以及 SPEA2 进行效果对比。

4.2 问题描述

多资源约束 (Multi-Resource Constrained, MRC) 下的并行机作业车间调度问题是对 SDST-PMJSS 问题的一个拓展，在原问题的基础上增加了多资源约束。多资源约束广泛存在于实际生产中，如现代汽车零部件加工，所有的加工资源组成一个加工中心，成型中通常要经过车、铣、刨、磨、钻等流程。在成型过程中不仅需要机械设备还需要刀具、夹具、冷却液、行车等辅助资源。加工设备可以看成机器约束，大型机器通常不能移动。而辅助资源一般比较昂贵，或者由于地方受限有些辅助资源体积较大，数量较少，不可能同时满足所有的加工需求，但是又必不可少。在工件到达相应的加工设备时，如果机器空闲而所需辅助资源缺乏不能满足加工要求会导致加工顺序紊乱，生产计划不能按时完成。为保证加工顺利进行，调度过程中必须考虑辅助资源的分配。由此可见，多资源约束问题比单资源约束问题复杂得多，也更符合现代智能制造企业。而单资源 SDST-PMJSS 问题已被证明为 NP-hard，所以多资源约束下的 SDST-PMJSS 问题同样是 NP-hard。因此，对该问题增加多资源约束在工程和理论上都具有一定意义。

多资源约束多目标 SDST-PMJSS 问题可定义为：在调度系统中存在 i 个工件，工件集合 $J = \{J_1, J_2, \dots, J_i\}$ ，所有工件需要在 k 台机器上加工，机器集合 $M = \{M_1, M_2, \dots, M_k\}$ ，每个工件包含多个工序，工序集合 $O = \{O_{i1}, O_{i2}, \dots, O_{ij}\}$ ，工序可选的加工机器数大于等于 1，可选加工机器集合 $M_{ij} \subseteq M$ 。工件加工过程中每道工序需要多种资源辅助加工，辅助资源集合 $R = \{R^1, R^2, \dots, R^l\}$ ， R^l 表示第 l 种资源，每种类型资源数量 g^l 有限并已知。工件需要在机器空闲并且所有辅

助资源到达才可以被加工，工件从机器 M_k 转换到机器 $M_{k'}$ 时存在一个序列依赖的准备时间，可以用来安装工具、调试设备等，具体描述详见节 2.1.3。因此，求解多资源约束下的多目标 SDST-PMJSS 问题可以分为以下三个子问题：

- (1) 各工件的机器选择 (Machine Selection, MS)。
- (2) 各机器上工序排序 (Operations Sequencing, OS)。
- (3) 辅助资源分配 (Resource Allocating, RA)。

在解决以上三个子问题的同时还需要考虑工件在机器间转换的准备时间，因此第三章中 SDST-PMJSS 问题的约束条件同样适用。此外考虑到资源分配，需要增加以下假设条件：

- (1) 所有资源均可重组且重复利用。
- (2) 所有资源在 0 时刻均可以使用，且上一机器所加工工序结束后，所释放资源可以立即被下一工序占用。
- (3) 同一资源在同一时间只能辅助一个工序加工。
- (4) 资源在被工序占用过程中不允许中断。

多资源约束并行机作业车间调度问题描述如图 4-1 所示。

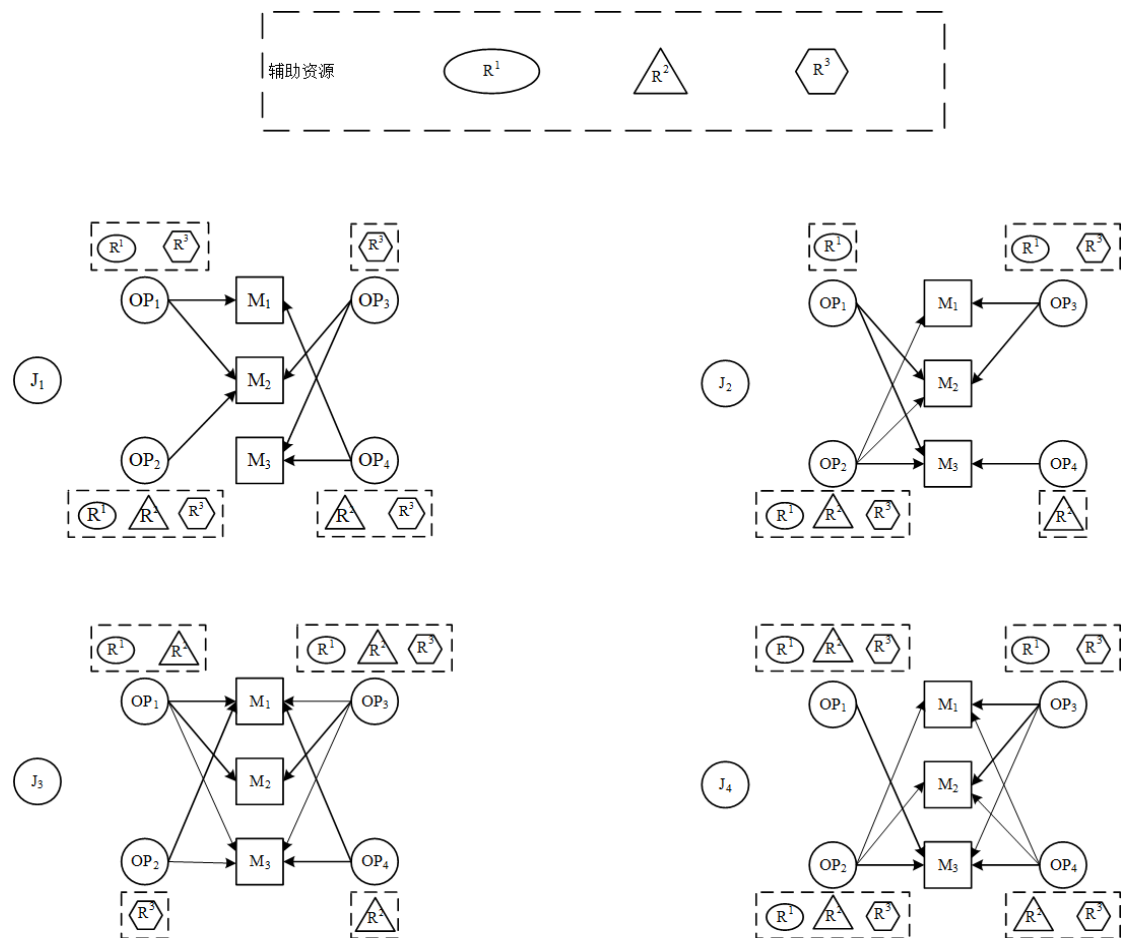


图 4-1 多资源约束并行机作业车间调度问题描述示意图

图 4-1 中 $\{R^1, R^2, R^3\}$ 表示三种不同的辅助资源, $\{J_1, J_2, J_3, J_4\}$ 表示待加工的四个工件, $\{OP_1, OP_2, OP_3, OP_4\}$ 工件的工序, $\{M_1, M_2, M_3\}$ 表示加工机器, 加工过程中各个工件的工序不仅需要机器还需要不同的辅助资源。

综上所述, 车间的加工能力不仅受机器的制约, 还受到其他辅助资源的约束, 机器空闲时, 工序所需的辅助资源不一定能满足加工要求。因此, 对多资源约束下 SDST-PMJSS 问题的研究需要在解决 MS 和 OS 两个子问题的同时考虑资源分配和序列依赖准备时间这两个条件对工序顺序的影响, 使调度目标达到最优。

4.3 多资源约束多目标 SDST-PMJSS 问题模型构建

4.3.1 相关变量和参数定义

由于增加了资源约束这一条件并且序列依赖准备时间类型和第三章有所不同, 因此需要对节 3.2.2 给出的相关变量和参数定义进行修改, 如下所示:

R	全部资源集合
l	资源类别序号, $l=1, 2, \dots, G$
R^l	第 l 种资源的集合
g^l	第 l 种资源总数
R_r^l	第 l 种资源的第 r 个资源, $r \in \{1, 2, \dots, g^l\}$
A_{ij}	工序 O_{ij} 所需的资源集合, $A_{ij} \in R$
B_{lij}	工序 O_{ij} 所需的第 l 种资源数
a_{ij}	加工工序 O_{ij} 的机器编号
$ST_{kk'}$	工件从机器 k 转换到机器 k' 的准备时间
S_{ij}	工序 O_{ij} 开始加工前的准备时间
NR_r^l	第 l 种资源的第 r 个资源辅助加工的工序总数
x_{ijqrl}	资源 l 的第 r 个资源在第 q 个单元分配给工序 O_{ij} 时为 1, 否则为 0

4.3.2 数学模型构建

目标函数:

$$f_1 = \min C_{\max} \quad (4-1)$$

$$f_2 = \min \sum_{i=1}^n d_i \quad (4-2)$$

$$f_3 = \sum_{i=1}^n \sum_{j=1}^{n_i} \sum_{k=1}^m \beta_{ijk} \cdot S_{ijk} \quad (4-3)$$

约束条件:

$$d_i \geq D_i - C_i \quad \forall i \quad (4-4)$$

$$JT_{ijk} + P_{ijk} + S_{ij} \leq C_{ijk} \quad \forall i, j, k \quad (4-5)$$

$$C_{ij} \leq JT_{i(j+1)} \quad \forall i, j \quad (4-6)$$

$$C_{in_i} \leq C_{max} \quad \forall i \quad (4-7)$$

$$a_{ij} = \sum_{k=1}^m \sum_{\lambda=1}^{N_k} k \cdot \beta_{ijk\lambda} \quad \forall i, j \quad (4-8)$$

$$S_{ij} = \begin{cases} ST_{a_{ij-1}, a_{ij}} & \forall i, j \in (1, n_i] \\ 0 & j=1 \end{cases} \quad (4-9)$$

$$JT_{ijk} - S_{ijk} = JS_{ijk} \quad (4-10)$$

$$\left(JT_{i'j'k} + P_{i'j'k} \right) \cdot \beta_{i'j'k\lambda} + S_{ij} \leq JT_{ijk} \cdot \beta_{ijk\lambda+1} + \varphi \left(\alpha_{(i'j')(ij)k}^{-1} \right) \quad (4-11)$$

$$i \neq i', j \neq j' \quad \forall i, i', j, j', \lambda, k$$

$$\left(JT_{ijk} + P_{ijk} \right) \cdot \beta_{ijk\lambda} + S_{i'j'} \leq JT_{i'j'k} \cdot \beta_{i'j'k\lambda+1} + \varphi \left(\alpha_{(i'j')(ij)k}^{-1} \right) \quad (4-12)$$

$$i \neq i', j \neq j' \quad \forall i, i', j, j', \lambda, k$$

$$\sum_{\lambda=1}^{N_k} \sum_{k=1}^m \beta_{ijk\lambda} = 1 \quad \forall i, j, \lambda, k \quad (4-13)$$

$$\sum_{i=1}^n \sum_{j=1}^{n_i} C_{ij} \cdot x_{ijqrl} \leq \sum_{i'=1}^n \sum_{j'=1}^{n_i} JT_{i'j'} \cdot x_{i'j'(q+1)rl} \quad i \neq i', j \neq j', q \in [1, NR_r^l - 1] \quad (4-14)$$

$$\sum_{q=1}^{NR_r^l} x_{ijqrl} \leq 1 \quad \forall i, j, r \quad \forall l \in [1, G] \quad (4-15)$$

$$\sum_{r=1}^{g^l} \sum_{q=1}^{NR_r^l} x_{ijqrl} \geq B_{lij} \quad \forall i, j \quad \forall l \in A_{ij} \quad (4-16)$$

$$JT_{ij}, S_{ij}, P_{ij}, d_i \geq 0 \quad \forall i, j \quad (4-17)$$

其中, 式(4-1)、式(4-2)和式(4-3)描述的是三个本文研究的目标函数, 分别为最小化总完工时间, 最小化总延迟时间和最小化总准备时间。式(4-4)表示了交货期和工件完成时间的差。式(4-5)表示工序 O_{ij} 在机器 k 第 λ 个加工单元开始加工前的准备时间、开始加工时间和加工时间和不大于其完工时间。式(4-6)表示工件工序的加工顺序, 同一工件在开始加工下一道工序前必须完成当前加工工序。式(4-7)表示加工系统的最后一道工序的完工时间不大于总完工时间。式(4-8)表示获取加工相应工序机器的编号方式。式(4-9)表示多资源约束 SDST-PMJSS 问题中序列相关准备时间的计算方式。式(4-10)表示每个工件每道工序序列依赖准备时间结束后立即开始加工。式(4-11)和式(4-12)表示同一机器上相邻两个不同工件工序的加工顺序, 即同一机器上一个工件工序加工完成后需要在下一工件工序准备完成才能开始加工。式(4-13)表示同一机器同一时间只能加工一个工件。式(4-14)表示占用有相同资源的工序需要在所占用的所有资源的前一个工序完成后才能开始加工。式(4-15)表示同一个资源在同一时间只能辅助加工一个工序。式(4-16)表示资源需要满足工序的加工需求。式(4-17)表示各工序的开始时间、加工时间、序列依赖准备时间以及工件的延迟时间均为正数。

4.4 混合 NSGA2 求解多资源约束多目标 SDST-PMJSS 问题

4.4.1 种群初始化

通过与第三章多目标 SDST-PMJSS 问题的描述和模型对比可以发现两个问题有相似之处, 但是本章问题的序列依赖准备时间体现在工件在机器转换之间, 并且增加了工序需要多种辅助资源加工这一约束, 使问题复杂度变大。

本章算法将沿用第三章种群初始化的方式, 即随机初始化方式和最小加工时间初始化方式按照 1:1 混合进行。NSGA2 部分的交叉变异方式和 MOABC 部分采蜜蜂混合搜索策略方式均保持不变。设计相应的解码策略来处理序列依赖准备时间和给各工序分配辅助资源。

4.4.2 个体的编码

由于多资源约束多目标 SDST-PMJSS 问题由各工件的机器选择、各机器上工序排序和辅助资源分配三个子问题组成。在解决方式上有集成式和分级式, 如 Afzalirad 和 Rezaeian^[90]在解决资源约束序列依赖准备时间不相关并行机调度问题时采用三重编码方式, 但是这种编码方式比较复杂, 染色体中所含信息比较多,

在解码过程中需要修正，降低算法的效率。如果采用分级式策略进行求解，编码部分仅考虑机器选择和工序排序，可以降低编码复杂度。

因此，本章结合第三章处理多目标 SDST-PMJSS 问题的方式，先确定机器选择和工序排序两个子问题。在个体解码部分再根据相应的解码策略对 SDST 和资源分配进行处理，确定每个工序的开始加工时间。在个体编码时采用分段式编码，完整的编码由机器选择部分 MS 和工序排序部分 OS 组成。具体详述见节 3.3.3。

4.4.3 两阶段解码策略

为降低编码复杂度，缩小搜索空间，提高算法效率，在编码过程中采用分段式编码，只考虑了机器选择和工序排序两个子问题。因此，在本节提出两阶段解码策略来确定工序的序列开始准备时间、辅助资源分配和工序开始加工时间。

根据节 4.2 对多资源约束的描述可知，工序在获得相应资源开始加工后不允许中断，则资源的被占用时间等于工序的加工时间。工序结束加工后，所占用资源立即被释放，并且被下一工序占用，则资源的开始被占用时间等于工序的开始加工时间。工序在机器间切换产生准备时间，因此准备时间与工序在工件上的紧前工序加工位置有关。由于工序一旦获取所有辅助资源立即开始加工，则序列依赖开始准备时间发生在等待辅助资源的过程中。所以一道工序开始加工必须同时满足以下四个条件：

- (1) 工件到达指定加工机器。
- (2) 工件要求的加工机器可用。
- (3) 工件获得全部的辅助资源。
- (4) 机器完成准备过程，即工序在机器切换序列依赖准备时间完成。

在获取所有加工信息后，解码过程分为两个阶段。第一阶段将机器视为加工资源，给各工序分配辅助资源，具体策略如下。

建立资源信息集合 RW 用于记录包含机器在内的所有资源的状态时间， $RW = \{\pi_k, \pi_R, \pi_{R_r^l}\}$ ，其中 π_k 记录机器 k 的可用时间； π_R 记录工序 O_{ij} 所需全部辅助资源的可用时间； $\pi_{R_r^l}$ 记录辅助资源 R_r^l 最早可用时间。工序 O_{ij} 在机器 k 上开始加工时间 JT_{ijk} ，加工时间 P_{ijk} 。

第一阶段先不考虑序列依赖准备时间，若要满足开始加工条件，工序 O_{ij} 的开始加工时间 JT_{ijk} 为：

$$JT_{ijk} = \max \left\{ JT_{i(j-1)k'} + P_{i(j-1)k'}, \pi_k, \pi_R \right\} \quad k \neq k' \quad (4-18)$$

当工序 O_{ij} 是工件 i 的第一道工序，即 $j=1$ 时， $JT_{i(j-1)k'} + P_{i(j-1)k'} = 0$ ， $JT_{ijk} = \max \{\pi_k, \pi_R\}$ 。

若工序 O_{ij} 需要从 G 种资源中分别选取一个资源作为辅助资源，则所有必须辅助资源最早可用时间为：

$$\pi_R = \max_{l \in \{1,2,\dots,G\}} \left\{ \min_{r \in \{1,2,\dots,g^l\}} \left\{ \pi_{R_r^l} \right\} \right\} \quad (4-19)$$

式(4-19)表示工序 O_{ij} 需要等到其所需辅助资源全部释放才能占用，如果资源 R_r^l 没有辅助任何工序加工，则 $\pi_{R_r^l} = 0$ ，工序 O_{ij} 可立即占用。

当工序 O_{ij} 完成加工后，机器 k 和所占用辅助资源立即被释放，按式(4-20)更新机器 k 的可用时间 π_k 和所有被占用辅助资源的可用时间 π_R 。

$$\pi_k = \pi_R = JT_{ijk} + P_{ijk} \quad (4-20)$$

在辅助资源分配过程中，工序 O_{ij} 对先前加工工序所释放的相同资源可能有多种选择方式。为了降低因辅助资源分配造成机器可用时间过长，工序延迟加工的影响，按式(4-21)^[91]在 RW 选择各辅助资源可用时间最接近 JT_{ijk} ，且小于 JT_{ijk} 的辅助资源。

$$R_r^l = \arg \left(\min_{R_r^l \in \left\{ l \mid l \in A_{ij}, JT_{ijk} - \pi_{R_r^l} \geq 0 \right\}} \left(JT_{ijk} - \pi_{R_r^l} \right) \right) \quad (4-21)$$

在加工开始，所有资源的可用时间均为 0，在每次辅助资源被选择和释放过程中都需要更新一次 RW ，按照上述资源分配策略从 OS 部分编码从左向右依次进行第一阶段解码，即可确定辅助资源分配后各工序的开始加工时间。

第二阶段解码是辅助资源全部分配结束后，在安排工件开始加工前要确定序列依赖准备时间。当机器在加工所有工件的第一道工序时不安排序列依赖时间，若机器上连续两个加工位置上的工序属于同一个工件，同样不安排序列依赖准备时间。

根据编码中的机器号，将每个工件的机器顺序和对应工序分出，从准备时间矩阵中获得序列依赖准备时间 S_{ij} 。如果工件在机器间转换，根据第一阶段解码获得的工序 O_{ijk} 开始加工时间 J_{ijk} ，计算该工序在相同机器上与其紧前工序 $O_{i'j'k}$ 的完工时间 $C_{i'j'k}$ 的差值 Δt 。按照下述规则计算工序 O_{ijk} 的开始准备时间并更新开始加工时间。

(1) 若 $\Delta t = 0$ ，则工序 O_{ijk} 的开始准备时间等于 $C_{i'j'k}$ ，工序开始加工时间调整为 $C_{i'j'k} + S_{ij}$ 。

(2) 若 $0 < \Delta t < S_{ij}$ ，则工序 O_{ijk} 开始准备时间等于 $C_{i'j'k} + \Delta t$ ，工序 O_{ijk} 开始加工时间调整为 $J_{ijk} + S_{ij}$ 。

(3) 若 $\Delta t \geq S_{ij}$ ，工序 O_{ijk} 开始准备时间等于 $J_{ijk} - S_{ij}$ ，工序 O_{ijk} 开始加工时间等于 J_{ijk} 。

下面给出一组 3×3 实例来说明两阶段解码策略在求解多资源约束下 SDST-PMJSS 问题的应用, 工件各工序的加工时间如表 4-1 所示; 工件在机器间转换的准备时间如表 4-2 所示; 各个工序所需资源如表 4-3 所示, 各类资源总数等于 2, 每个资源均使用整数 1 表示。

表 4-1 工件各工序的加工时间

工件	工序	机器		
		M_1	M_2	M_3
J_1	O_{11}	4	4	-
	O_{12}	-	8	-
	O_{21}	-	10	8
J_2	O_{22}	5	5	3
	O_{23}	7	9	-
J_3	O_{31}	8	-	6
	O_{32}	6	-	-

表 4-2 工件在机器转换过程中的序列依赖准备时间

机器	M_1	M_2	M_3
M_1	0	1	2
M_2	2	0	1
M_3	1	2	0

表 4-3 各工序所需资源

工件	工序	辅助资源		
		R^1	R^2	R^3
J_1	O_{11}	1	0	1
	O_{12}	0	0	1
	O_{21}	1	0	0
J_2	O_{22}	1	1	1
	O_{23}	1	0	1
J_3	O_{31}	1	1	0
	O_{32}	0	1	0

根据表 4-1 生成一个可行解编码如图 4-2 所示。

MS	2	2	3	3	2	1	1
OS	3	1	2	1	3	2	2

图 4-2 可行解编码

根据图 4-2 所示的可行解编码，下面详细介绍解码过程中辅助资源的分配和各个工序开始加工时间的确定。

第一阶段：OS 部分从左向右依次与 MS 部分对应解码，辅助资源库可以表示为：

$$R = \begin{pmatrix} R_1^1 & R_1^2 & R_1^3 \\ R_2^1 & R_2^2 & R_2^3 \end{pmatrix}$$

第一个工序 O_{31} 选择机器 M_1 ，工序 O_{31} 需要辅助资源 R^1 ， R^2 各一个，此时 $t=0$ ，机器和辅助资源均可以使用，则给工序 O_{31} 随机分配相应辅助资源 R_1^1 和 R_1^2 。由于加工过程中不允许中断，且 $P_{311}=8$ ，所以机器 M_1 和辅助资源 R_1^1 和 R_1^2 的可用时间更新为 8。第二个工序 O_{11} 选择机器 M_2 ，工序 O_{11} 需要辅助资源 R^1 和 R^3 各一个，此时机器和剩余辅助资源均可以使用，则工序 O_{11} 占用辅助资源 R_2^1 和 R_1^3 ， $P_{112}=4$ ，更新机器 M_2 ，辅助资源 R_2^1 和 R_1^3 可用时间为 4。第三个工序 O_{21} 选择机器 M_3 ，需要一个辅助资源 R^1 ，此时机器空闲，但是所有辅助资源 R^1 被占用，因此工序 O_{21} 无法加工。辅助资源 R_1^1 和 R_2^1 可用时间分别为 4 和 8，因此工序 O_{21} 选择可用时间最小的辅助资源 R_1^1 ，即当 $t=4$ 时，工序 O_{21} 满足开始加工条件，开始加工， $P_{212}=8$ ，更新机器 M_3 和辅助资源 R_1^1 可用时间 $\pi_3=\pi_{R_1^1}=4+8=12$ 。第四个工序 O_{12} 选择机器 M_2 ，需要一个辅助资源 R^3 。由于工序 O_{12} 与工序 O_{11} 在同一机器上加工，且当 $t=4$ 时，辅助资源 R_1^3 被释放，所以工序 O_{12} 可在此时刻随机挑选辅助资源 R_1^3 和 R_2^3 开始加工，同时更新机器和所占用辅助资源可用时间为 12。以此类推，直至所有工序被分配并确定开始加工时间。

第二阶段：将工序分组，确定序列依赖准备时间，调整各工序开始加工时间。

工件 J_1 各工序加工机器顺序为： $(O_{11}, M_2) \rightarrow (O_{12}, M_2)$ ；工件 J_2 各工序加工机器顺序为： $(O_{21}, M_3) \rightarrow (O_{22}, M_3) \rightarrow (O_{23}, M_1)$ ；工件 J_3 各工序加工机器顺序为： $(O_{31}, M_1) \rightarrow (O_{32}, M_1)$ 。由此可见，该示例只有工件 J_2 在加工工序 O_{23} 时工件

从机器 M_3 转换到机器 M_1 ，从表 4-2 中获得序列依赖准备时间 $S_{23}=1$ 。工序 O_{23} 在机器 M_1 上紧前工序 O_{32} 完工时间 $C_{321}=14$ ，工序 O_{23} 开始加工时间 $J_{231}=15$ 。 $\Delta t = J_{231} - C_{231} = 1$ ， $\Delta t = S_{23}$ ，根据第二阶段规则，工序 O_{23} 的序列依赖开始准备时间等于 $J_{231} - S_{23} = 14$ ，工序 O_{23} 最终开始加工时间 $J_{231} = 15$ 。据上述生成的 3×3 调度问题示例可行解甘特图如图 4-3 所示。

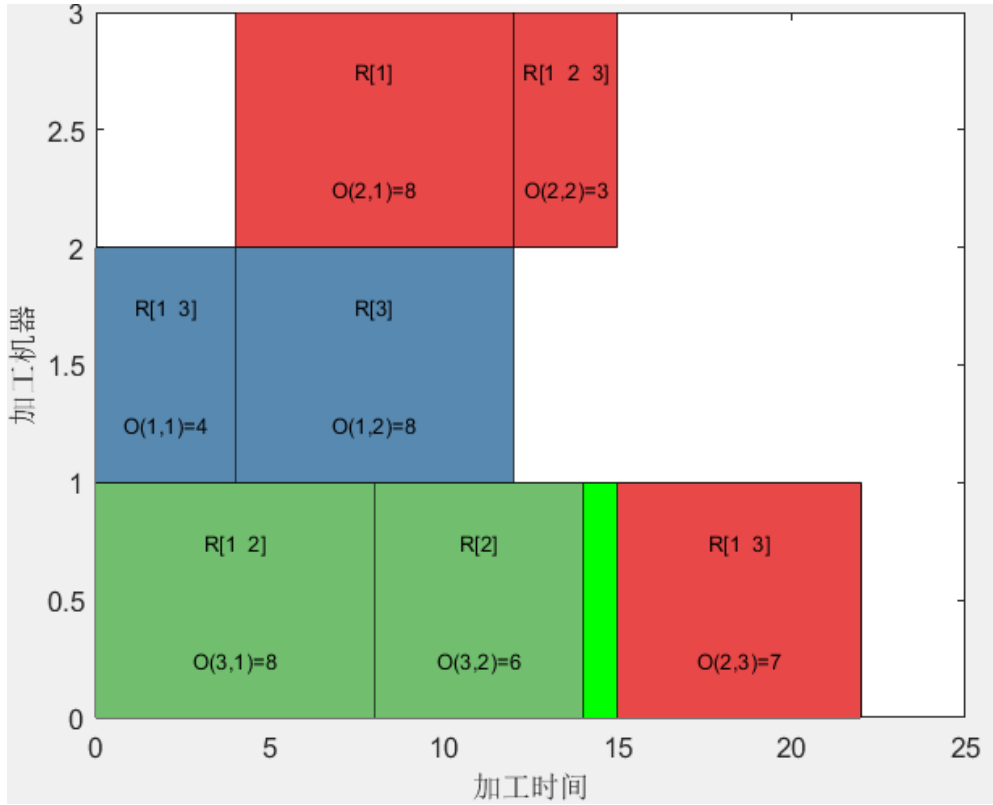


图 4-3 3×3 调度问题示例可行解甘特图

4.5 实验设计与结果分析

4.5.1 仿真环境

本章混合 NSGA2 通过 MATLAB 语言编程实现，仿真软件为 MATLAB R2019b。运行环境为：AMD Ryzen 5 4600U CPU，主频为 2Ghz，内存为 16G，操作系统为 Windows 10。

4.5.2 实验数据设计

本节测试数据参考文献^[92]生成 7 组每组 3 种数据集分别以 {1,2,3} 表示，以验证混合 NSGA2 对各种问题规模的适应性。在这些测试数据中，每组工件加工

时间采用均匀分布产生, 区间分别为 $\{1,2,\dots,15\}$, $\{1,2,\dots,30\}$, $\{1,2,\dots,50\}$ 。序列依赖准备时间同样采用均匀分布产生, 区间为 $\{1,2,\dots,5\}$ 。设定三种辅助资源 $\{R^1, R^2, R^3\}$, 每种资源总数分别为 $\{18,16,20\}$, 每个工序需要0~3种资源各一个。随机产生的问题包含机器数 $m = \{10,20,30,40\}$, 工件数 $n = \{60,100,150,200\}$, 每个工件的工序数区间为 $\{1,2,\dots,5\}$ 。如此, 这些随机测试实例范围广, 规模比较大。序列依赖准备时间部分实例如表 4-4 所示。

表 4-4 工件在机器间转换的序列依赖准备时间部分示例

机器	M_1	M_2	M_3
M_1	0	5	1
M_2	3	0	3
M_3	2	2	0

4.5.3 参数设置

本章实验在对比算法上使用目前求解多目标问题效果比较突出的 NSGA2、MOABC、改进强度的 Pareto 算法 (SPEA2) 三种算法基本形式。三种算法的相同参数如种群规模、迭代次数、交叉率与变异率等设置相同。混合 NSGA2 具体设置如表 4-4 所示。NSGA2 与 MOABC 具体参数设置与节 3.6.3 相同, SPEA2 算法具体参数设置参考文献^[93]。

4-5 混合 NSGA2 参数设置

种群规模	交叉率 P_c	变异率 P_m	$maxCycle$	外部存档容量	$limit$
60	0.9	0.1	200	30	20

4.5.4 实验结果与分析

为了验证混合 NSGA2 在求解多资源约束多目标 SDST-PMJSS 问题的性能, 在实验中算法采用相同的编码、解码方案以及辅助资源。每个实验例子在四个算法上均独立运行 20 次, 各算法的参数均统一设置, 以保证算法对比的公平性, 降低随机误差的影响。每组数据在四个算法上将一共得到 80 组 Pareto 解集, 将各组的解集通过节 2.6.2 给出的三个评价指标计算方式得出的三个指标值, 作为最终的实验结果, 以对比各算法求解该问题的效果, 为简化数据形式, MID 和 SNS 值均除以 10^5 。四种算法在多目标评价指标上的实验结果如表 4-6 和表 4-7 所示。

表 4-6 NSGA2 和 MOABC 在三种多目标评价指标上的实验结果

$n \times m$		NSGA2			MOABC		
		MID	SNS	POD	MID	SNS	POD
1	60×10	2.4326	2.5124	5.00	2.4325	2.5100	5.00
2		2.4330	2.5372	7.69	2.4330	2.5368	19.23
3		2.4338	2.5384	16.00	2.4338	2.5380	16.00
1	100×20	2.4298	2.5153	4.12	2.4297	2.5130	4.12
2		2.4296	2.5259	7.69	2.4294	2.5273	7.69
3		2.4370	2.3773	5.00	2.4373	2.3746	15.00
1	60×40	2.4327	2.5267	7.69	2.4326	2.5273	7.69
2		2.4325	2.5358	20.00	2.4325	2.5351	20.00
3		2.4323	2.5346	9.76	2.4329	2.5351	9.76
1	100×40	2.4308	2.5563	9.76	2.4307	2.555	9.76
2		2.4311	2.5307	7.69	2.4310	2.5241	7.69
3		2.4306	2.5031	5.00	2.4317	2.5055	25.00
1	150×30	2.4274	2.5498	5.00	2.4271	2.5481	5.00
2		2.4276	2.5020	9.76	2.4277	2.4945	9.76
3		2.4340	2.3702	12.50	2.4325	2.3902	25.00
1	150×40	2.4273	2.5457	4.12	2.4270	2.5434	4.12
2		2.4269	2.5404	16.92	2.4266	2.5412	6.15
3		2.4267	2.5400	25.00	2.4267	2.5383	15.00
1	200×40	2.4242	2.5146	6.15	2.4238	2.5252	6.15
2		2.4242	2.5185	5.00	2.4241	2.5087	5.00
3		2.4239	2.5241	7.69	2.4239	2.5156	7.69

表 4-7 SPEA2 和混合 NSGA2 在三种多目标评价指标上的实验结果

$n \times m$		SPEA2			混合 NSGA2		
		MID	SNS	POD	MID	SNS	POD
1	60×10	2.4326	2.5122	15.00	2.4328	2.5133	75.00
2		2.4330	2.5368	19.23	2.4331	2.5384	65.38
3		2.4338	2.5386	16.00	2.4340	2.5393	52.00
1	100×20	2.4297	2.5235	4.12	2.4300	2.5185	87.63
2		2.4296	2.5292	7.69	2.4300	2.5315	76.92
3		2.4364	2.3842	5.00	2.4334	2.4390	75.00
1	60×40	2.4329	2.5289	30.77	2.4331	2.5295	53.85
2		2.4327	2.5367	20.00	2.4332	2.5384	40.00
3		2.4330	2.5361	36.34	2.4333	2.5376	44.15
1	100×40	2.4308	2.5570	9.76	2.4315	2.5587	70.73
2		2.4314	2.5319	19.23	2.4313	2.5484	65.38
3		2.4310	2.5064	5.00	2.4313	2.5313	65.00
1	150×30	2.4282	2.5508	75.00	2.4282	2.5508	75.00
2		2.4276	2.5106	34.15	2.4278	2.5137	46.34
3		2.4316	2.4256	25.00	2.4320	2.4028	37.50
1	150×40	2.4272	2.5351	22.68	2.4276	2.5499	69.07
2		2.4267	2.5463	6.15	2.4270	2.5475	70.77
3		2.4266	2.5443	5.00	2.4269	2.5463	55.00
1	200×40	2.4239	2.5334	6.15	2.4242	2.5366	81.54
2		2.4241	2.5183	5.00	2.4243	2.5328	85.00
3		2.4239	2.5272	7.69	2.4242	2.5380	76.92

本章混合 NSGA2 在 60×10 第三组算例上求得的最优调度甘特图如图 4-3 所示。从图 4-3 中可知，在 0 时刻加工的工件所需辅助资源满足其加工需求，因此在 0 时刻所有分配到工件的机器可以同时加工。机器 M_2 前两个加工位置分别加工工件 J_{38} 和 J_{15} 的第一个工序，因此没有序列依赖准备时间。在机器 M_2 上第三个加工位置加工工件 J_{47} 的第二个工序 O_{472} ，而按照工艺要求，其紧前工序 O_{471} 被分配到机器 M_{10} 的第二个加工位置，则工件 J_{47} 的前两个工序在加工过程中从机器 M_{10} 转换到了机器 M_2 ，工序 O_{472} 开始加工前有序列依赖准备时间，即 $S_{472}=2$ 。在机器 M_{10} 上，工序 O_{471} 的加工时间等于 9，完工时间等于 18，由于工序 O_{472} 所需辅助资源满足其加工要求，所以工序 O_{472} 的序列依赖开始准备时

间等于 16，开始加工时间等于 18。

根据表 4-6 和表 4-7 中本章的混合 NSGA2 与 NSGA2、MOABC 和 SPEA2 在三个多目标评价指标的结果，可以得出以下结论：

（1）从实验结果中可以看出，21 组数据中混合 NSGA2 的 SNS 值均高于其他三种算法，说明本章混合算法能够有效求解多资源约束多目标 SDST-PMJSS 问题，产生的 Pareto 解集多样性较好。

（2）从 POD 指标上来看，混合 NSGA2 在 21 组数据中的 POD 值均超过其他三种对比算法，尤其在第 1、2、4、5 等共 14 组数据中 POD 值均超过了 60%，说明该算法产生的 Pareto 解集相对于其他三种算法支配性更强，所产生解的质量更好，算法综合性能更优。

（3）从 MID 指标上来看，本章混合 NSGA2 值略大于其他三种对比算法，说明混合 NSGA2 的收敛速度略慢于其他三种算法，但差距很小。因为混合 NSGA2 在交叉、变异产生子代后又使用蜂群对每一个子代进一步优化，产生新解的可能性更大。问题复杂度提高，解空间变大时，最优解被重复搜索和蜂群优化阶段中劣解随机产生的可能性会相对变大。

综上所述，针对本章问题，将混合 NSGA2 和 NSGA2、MOABC、SPEA2 三种在多目标问题上表现较好的算法，在 7 大组共 21 小组不同大规模的数据上测试对比。所得实验结果说明混合 NSGA2 在求解多资源约束并且序列依赖准备时间发生于工件在机器上转换产生的问题相对于其他三种算法更优越，算法性能更好。

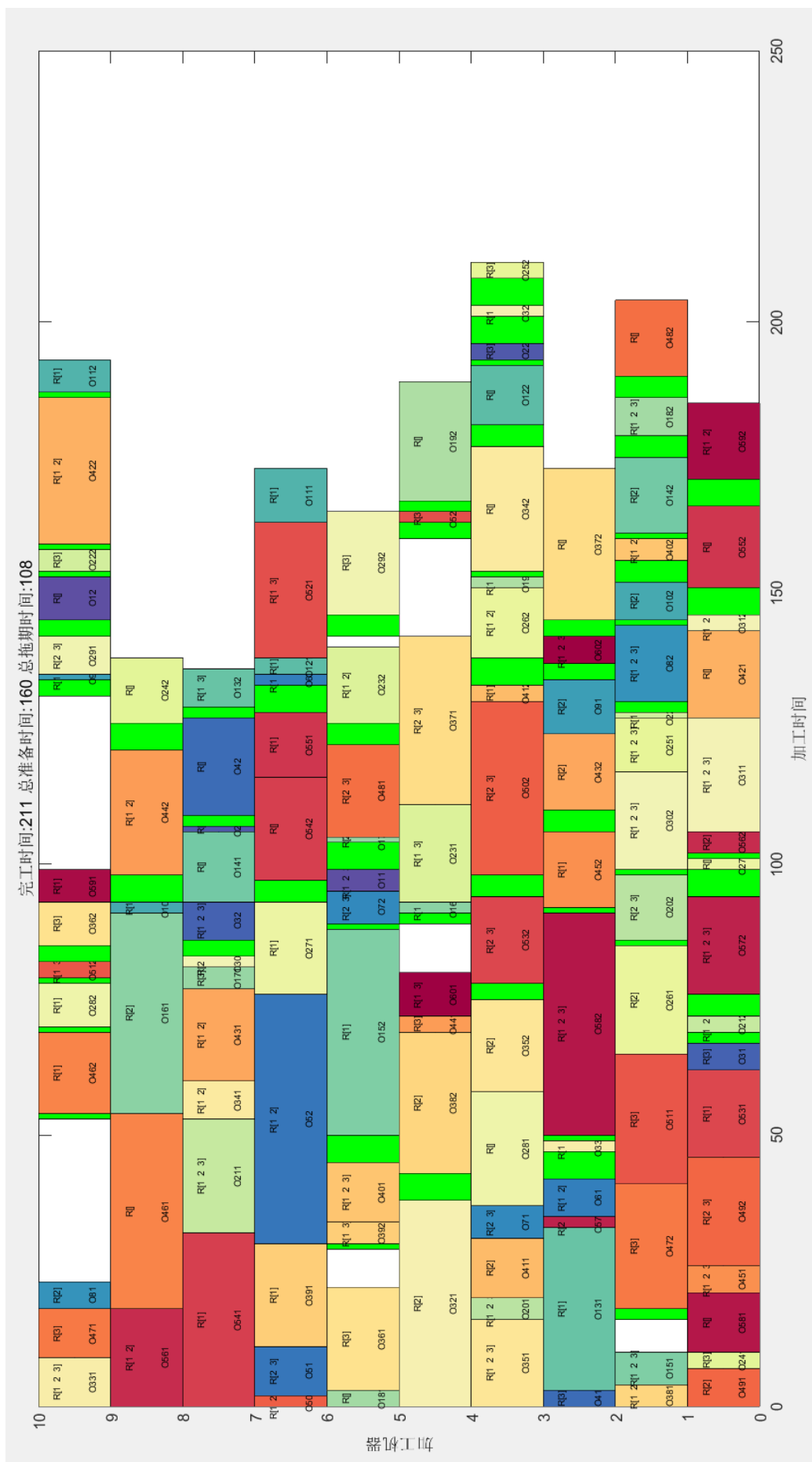


图 4-4 混合 NSGA2 在 60×10 第三组算例上的最优调度甘特图

4.6 本章小结

本章求解了多资源约束多目标 SDST-PMJSS 问题, 该 SDST 产生于工件在不同机器间转换的过程中。首先, 对多资源约束调度问题在智能制造车间的应用做出详述; 然后, 沿用第三章提出的最小化总完工时间、最小化总延迟时间和最小化总准备时间三个目标, 建立了多资源约束多目标 SDST-PMJSS 调度问题的数学模型; 接着, 根据问题特点采用分段式编码, 将资源分配和 SDST 问题放在解码策略中解决。设计了两阶段解码策略以确定各工序的开始加工时间, 第一阶段根据编码方案得到第一次各工序在没有分配资源时的开始加工时间, 将机器视为加工资源, 和辅助资源一起建立资源集合以记录各个资源的加工信息, 根据各工序所需的资源进行分配, 得到资源分配后各工序的第二次开始加工时间; 第二阶段根据机器上的加工顺序以及工件工艺流程, 确定机器开始准备时间和各工序的第三次开始加工时间。经过三次开始加工时间的确定就能得到整个加工系统的各项信息, 进行有效调度。最后, 参考文献中算例生成的方式, 生成不同规模的算例, 通过本章混合 NSGA2 与 NSGA2、MOABC 和 SPEA2 在各种算例上得到的 Pareto 解集计算出三种多目标评价指标值, 验证了本章提出的混合 NSGA2 在求解多资源约束多目标 SDST-PMJSS 问题时具有显著的优越性。

研究总结与展望

（一）研究总结

首先，本文针对序列依赖准备时间产生于同一机器不同工件之间的多目标 SDST-PMJSS 问题，提出混合 NSGA2 算法。然后，改进混合 NSGA2，提出两阶段解码策略，求解了序列依赖准备时间产生于相同工件在不同机器间转换的多资源约束多目标 SDST-PMJSS 问题。本文的主要工作总结如下：

（1）详细描述了 SDST-PMJSS 问题的理论、特点以及应用背景。首先对准备时间进行分类，选择同一机器不同工件之间和不同机器相同工件之间两种序列依赖准备时间进行讨论。然后详述了 NSGA2 和 ABC 算法的核心思想、基本原理以及算法流程，对两种算法的优缺点进行归纳总结，为提出带有人工蜂群搜索策略的混合 NSGA2 提供理论依据。

（2）在相同机器上加工不同工件过程中产生 SDST 的情形下，研究了多目标 SDST-PMJSS 问题以及解决方法。首先，建立了最小化总完工时间、最小化总延迟时间和最小化总准备时间的多目标 SDST-PMJSS 数学模型。然后，根据 Pareto 支配概念将 ABC 算法修改为 MOABC 算法；将 MOABC 算法中采蜜蜂、跟随蜂与侦查蜂引入 NSGA2 中，利用 MOABC 中采蜜蜂和跟随蜂邻域搜索以及侦查蜂随机产生解的特点对 NSGA2 父代种群交叉变异后产生的子代进一步优化，对每一个个体进行邻域搜索，并进行贪婪选择，提高局部搜索能力，提出了带有人工蜂群搜索策略的混合 NSGA2。最后，通过与传统 NSGA2 和 MOABC 对比，验证了混合 NSGA2 在求解该问题时相对其他两种算法在算法支配性、解的多样性以及解的质量上有着更大的优势。

（3）在加工过程中相同工件在不同机器间转换时产生 SDST 的情形下，研究了多目标 SDST-PMJSS 问题，并且增加了多资源约束这一条件以更符合实际智能制造。首先，沿用了最小化总完工时间、最小化总延迟时间和最小化总准备时间三个目标函数建立了多资源约束多目标的 SDST-PMJSS 问题数学模型。然后，采用分段式编码方式，提出了两阶段解码策略。在解码中将机器与三种辅助资源一起列为加工资源，建立资源信息集合，根据编码获得各工序的加工顺序，通过建立的集合中信息的更新给各工序分配资源。解码第二阶段获取各工序的 SDST，以确定所有工序的开始加工时间。最后，通过与传统 NSGA2、MOABC 和 SPEA2 对比，验证了该算法在求解多资源约束多目标 SDST-PMJSS 的有效性与优越性。

（二）研究展望

本文对两种序列依赖准备时间的多目标多并行机作业车间调度问题进行了研究，并且在第二种问题中增加了多资源约束的限制条件，提出了带有人工蜂群搜索策略的混合 NSGA2，但仍存在以下不足：

（1）本文所提的混合 NSGA2 是用人工蜂群搜索的邻域搜索特点来进一步优化交叉变异产生的子代，虽然能够获得更优质的解，并提高了 NSGA2 的局部搜索能力，但是算法收敛速度有所下降。因此，未来工作可以在蜂群搜索阶段设计相应的搜索策略引导蜂群向更优质的解的方向搜索，减小劣解产生的可能性。

（2）本文研究的均属于静态的调度问题，而实际的调度环境是动态的，很多事情是随机的，并且现在环境保护非常重要，也是一个研究热点。因此，未来工作可以从动态和绿色调度问题着手研究。

（3）调度理论产生于实际问题因此也要应用到实际生产中，为适应当前的迅速发展的智能制造，可以开发具有比较成熟的调度算法的信息化软件，将成功的调度技术运用到实际生产中，产生经济效益。

参考文献

- [1] Potts, C N, Strusevich, V A. Fifty years of scheduling: A survey of milestones[J]. Journal of the Operational Research Society, 2009, 60(1):41–68.
- [2] Graham R L, Lawler E L, Lenstra J K, et al. Optimization and approximation in deterministic sequencing and scheduling: A survey[J]. Annals of Discrete Mathematics, 1979, 5(1):287-326.
- [3] Kopanos G M, Miguel M, Puigjaner L. An efficient mixed-integer linear programming scheduling framework for addressing sequence-dependent setup issues in batch plants[J]. Industrial & Engineering Chemistry Research, 2009, 48(13):6346-6357.
- [4] Johnson S. Optimal two and three stage production schedules with times include[J]. Naval Research Logistics Quarterly, 1954, 1:61-68.
- [5] Ou J, Zhong X, Wang G. An improved heuristic for parallel machine scheduling with rejection[J]. European Journal of Operational Research, 2015, 241(3):653-661.
- [6] Mensendiek A, Gupta J, Herrmann J. Scheduling identical parallel machines with fixed delivery dates to minimize total tardiness[J]. European Journal of Operational Research, 2015, 243(2):514-522.
- [7] Nattaf M, Artigues C, Lopez P, et al. A batch sizing and scheduling problem on parallel machines with different speeds, maintenance operations, setup times and energy costs[C]// 2015 International Conference on Industrial Engineering and Systems Management (IESM). IEEE, 2016.
- [8] Cota L P, Guimaraes F G, Oliveira F, et al. An Adaptive Large Neighborhood Search with Learning Automata for the Unrelated Parallel Machine Scheduling Problem[C]// 2017 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2017.
- [9] Zhang R, Song S, Wu C. A hybrid artificial bee colony algorithm for the job shop scheduling problem[J]. International Journal of Production Economics, 2013, 141(1):167-178.
- [10] Chen J C, Wu C C, Chen C W, et al. Flexible job shop scheduling with parallel machines using Genetic Algorithm and Grouping Genetic Algorithm[J]. Expert Systems with Applications An International Journal, 2012, 39(11):10016-10021.
- [11] Chiang T C, Lin H J. A simple and effective evolutionary algorithm for multi objective flexible job shop scheduling[J]. International Journal of Production Economics, 2013, 141(1):87-98.
- [12] Pinson J C. An Algorithm for Solving the Job-Shop Problem[J]. Management Science, 1989, 35(2):164-176.
- [13] Oezgüven C, Yavuz Y, Oezbakir L. Mixed integer goal programming models for the flexible job-shop scheduling problems with separable and non-separable sequence dependent setup times[J]. Applied Mathematical Modelling, 2012, 36(2):846-858.

- [14] Hansmann R S, Rieger T, Zimmermann U T. Flexible job shop scheduling with blockages[J]. *Mathematical Methods of Operations Research*, 2014, 79(2):135-161.
- [15] Chen H, Luh P B. An alternative framework to Lagrangian relaxation approach for job shop scheduling[J]. *European Journal of Operational Research*, 2003, 149(3):499-512.
- [16] Adams J, Zawack B D. The shifting bottleneck procedure for job shop scheduling[J]. *Management Science*, 1988, 34(3):391-401.
- [17] Liu S Q, Kozan E. A hybrid shifting bottleneck procedure algorithm for the parallel-machine job-shop scheduling problem[J]. *Journal of the Operational Research Society*, 2012, 63(2), 168-182.
- [18] Odili J B, Kahar M N M, Noraziah A. Swarm intelligence optimization algorithms: A review[J]. *Telecommunication. Electronic Computer. Engineering*, 2018, 10(4), 139–142.
- [19] Suresh L P, Dash S S, Panigrahi B K. Artificial intelligence and evolutionary algorithms in engineering systems[J]. *Advances in Intelligent Systems & Computing*, 2015, 325(10):643-648.
- [20] Chaudhry A, Elbadawi I. Minimisation of total tardiness for identical parallel machine scheduling using genetic algorithm[J]. *Sādhanā*, 2017, 42(1):11-21.
- [21] Lausch S, Mönch L. Metaheuristic approaches for scheduling jobs on parallel batch processing machines[J]. *Journal of the Operational Research Society*, 2016, 236(2):186-207.
- [22] Rajkanth R, Rajendran C, Ziegler H. Heuristics to minimize the completion time variance of jobs on a single machine and on identical parallel machines[J]. *The International Journal of Advanced Manufacturing Technology*, 2017.88(7):1923-1936.
- [23] Hamzadayi A, Yildiz G. Modeling and solving static m identical parallel machines scheduling problem with a common server and sequence dependent setup times[J]. *Computers & Industrial Engineering*.2017.106(2):287–298.
- [24] Aalaei A, Kayvanfar V, Davoudpour H. A multi-objective optimization for preemptive identical parallel machines scheduling problem[J]. *Computational & Applied Mathematics*, 2015, 36:1-21.
- [25] Kayvanfar V, Zandieh M, Teymourian E. An intelligent water drop algorithm to identical parallel machine scheduling with controllable processing times: A just-in-time approach[J]. *Computational & Applied Mathematics*, 2017, 36(1):159-184.
- [26] Wang S, Wang X, Yu J, et al. Bi-objective identical parallel machine scheduling to minimize total energy consumption and makespan[J]. *Journal of Cleaner Production*, 2018, 193(5):424-440.
- [27] An Y J, Kim Y D, Choi S W. Minimizing makespan in a two-machine flowshop with a limited waiting time constraint and sequence-dependent setup times[J]. *Computers & Operations Research*, 2016, 71(1):127-136.

- [28] Ince Y, Karabulut K, Tasgetiren M F, et al. A discrete artificial bee colony algorithm for the permutation flowshop scheduling problem with sequence-dependent setup times[C]// 2016 IEEE Congress on Evolutionary Computation. IEEE, 2016.
- [29] Harbaoui H, Bellenguez O, Khalfallah S, Scheduling a two-stage hybrid flow shop with dedicated machines, time lags and sequence-dependent family setup times[C]// 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 2016.
- [30] Ahonen H, Alvarenga A D. Scheduling flexible flow shop with recirculation and machine sequence-dependent processing times: formulation and solution procedures[J]. International Journal of Advanced Manufacturing Technology, 2017, 89(3):765-777.
- [31] Zhang Z Q, Qian B, Liu B, et al. Hybrid estimation of distribution algorithm for blocking flow-Shop scheduling problem with sequence-dependent setup times[J]. Intelligent Computing Theories and Application, 2018, 10954(6):628-637.
- [32] Li X, Yang Z, Ruiz R, et al. An iterated greedy heuristic for no-wait flow shops with sequence dependent setup times, learning and forgetting effects[J]. Information Sciences, 2018, 453(4):408-425.
- [33] Azzouz A, Ennigrou M, Said L B. Flexible Job-shop Scheduling Problem with Sequence-dependent Setup Times using Genetic Algorithm[C]// 18th International Conference on Enterprise Information Systems. SCITEPRESS - Science and Technology Publications, Lda, 2016.
- [34] Sharma P, Jain A. New setup-oriented dispatching rules for a stochastic dynamic job shop manufacturing system with sequence-dependent setup times[J]. Concurrent engineering, 2016, 24(1):58-68.
- [35] Velez-Gallego M C, Maya J, Montoya Torres J R. A beam search heuristic for scheduling a single machine with release dates and sequence dependent setup times to minimize the makespan[J]. Computers & Operations Research, 2016, 73(4):132-140.
- [36] Yue L, Guan Z, Saif U, et al. Hybrid pareto artificial bee colony algorithm for multi-objective single machine group scheduling problem with sequence-dependent setup times and learning effects[J]. SpringerPlus, 2016, 5(1):1593-1624.
- [37] Wang L, Wang S, Zheng X. A hybrid estimation of distribution algorithm for unrelated parallel machine scheduling with sequence-dependent setup times[J]. Journal of Automatica Sinica, 2016. 3(3):235-246.
- [38] Shen L, Dauzère-Pérès, Stéphane, Neufeld J S. Solving the flexible job shop scheduling problem with sequence-dependent setup times[J]. European Journal of Operational Research, 2017, 265(8):503-516.
- [39] Bektur G, Saraç T. A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server[J]. Computers & Operations Research, 2019, 103(10):46-63.

- [40] Kim J G, Song S, Jeong B J. Minimising total tardiness for the identical parallel machine scheduling problem with splitting jobs and sequence-dependent setup times[J]. *International Journal of Production Research*, 2020, 58(6):1628-1643.
- [41] 胡大勇,姚振强.调整时间与顺序相关的等同并行机调度[J].*机械工程报*,2011,47(16):160-165.
- [42] 徐建有,董乃群,顾树生.带有顺序相关调整时间的多目标流水车间调度问题[J].*计算机集成制造系统*,2013,19(12):3170-3176.
- [43] 李冰,黎展滔.调整时间与顺序相关的能耗优化调度问题[J].*机电工程*,2015,44(06):12-16.
- [44] 查靛,金花,潘志成,等.基于顺序相关调整时间的 FJSP 与设备维护计划集成优化[J].*组合机床与自动化加工技术*,2016(05):155-160.
- [45] 杨武成, 程文明. 求解考虑顺序相关调整时间的双边装配线平衡问题的变邻域搜索算法[J]. *工业工程*, 2019(5):68-74.
- [46] Afzalirad M, Rezaeian J. Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions[J]. *Computers & Industrial Engineering*, 2016, 98(5):40-52.
- [47] Zheng X L, Wang L. A knowledge-guided fruit fly optimization algorithm for dual resource constrained flexible job-shop scheduling problem[J]. *International Journal of Production Research*, 2018, 54(18):5554-5566.
- [48] Yazdani M, Zandieh M, Tavakkoli-Moghaddam R, et al. Two meta-heuristic algorithms for the dual-resource constrained flexible job-shop scheduling problem[J]. *Scientia Iranica*, 2015, 22(3):1242-1257.
- [49] Wang S, Ming L. Multi-objective optimization of parallel machine scheduling integrated with multi-resources preventive maintenance planning[J]. *Journal of Manufacturing Systems*, 2015, 37(7):182-192.
- [50] Villa F, Vallada E, Fanjul-Peyro L. Heuristic algorithms for the unrelated parallel machine scheduling problem with one scarce additional resource[J]. *Expert Systems with Application*, 2018, 93(9):28-38.
- [51] Fu Y, Tian G, Li Z, et al. Parallel machine scheduling with dynamic resource allocation via a master-slave genetic algorithm[J]. *IEEE Transactions on Electrical and Electronic Engineering*, 2018, 13(5):748-756.
- [52] Zhong Y, Li J M, Zhu S Z. Research on the multi-objective optimized scheduling of the flexible job-Shop considering multi-Resource allocation[J]. *International Journal of Simulation Modelling*, 2017, 16(3):517-526.
- [53] Li J, Yuan H, Niu X. A branch population genetic algorithm for dual-resource constrained job shop scheduling problem[J]. *Computers & Industrial Engineering*, 2016, 102:113-131.
- [54] 李兢尧,黄媛,王军强.基于压缩时窗的双资源约束作业车间调度策略[J].*计算机集成制造系统*,2016,22(12):2827-2835.

- [55] 孙斌,吉卫喜,堵士俊,等.基于改进萤火虫算法的多资源多目标车间调度问题研究[J].现代制造工程,2016(02):65-72.
- [56] 陈少,吉卫喜,仇永涛,等.基于改进鸟群算法的双资源约束离散智能车间调度问题研究[J].现代制造工程,2019(04):20-26.
- [57] 娄航宇,张吉善,赵云博.基于扩展双资源约束型航空构件制造车间调度方法[J].计算机集成制造系统,2020,26(12):3329-3340.
- [58] 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[J].电子学报,2021,49(02):324-330.
- [59] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002,6(2):182-197.
- [60] Chen C, Ji Z, Wang Y. NSGA-II applied to dynamic flexible job shop scheduling problems with machine breakdown[J]. Modern Physics Letters B, 2019,32(34-36):1840411.
- [61] Gong X, Deng Q, Gong G, et al. A memetic algorithm for multi-objective flexible job-shop problem with worker flexibility[J]. International Journal of Production Research, 2018, 56(7):2506-2522.
- [62] Xu L, Xia Z Y, Ming H. Study on improving multi-objective flexible job shop scheduling based on Memetic algorithm in the NSGA-II framework[C]// International Conference on Cloud Computing & Internet of Things. IEEE, 2016,1-7.
- [63] 陈辅斌,李忠学,杨喜娟.基于改进 NSGA2算法的多目标柔性作业车间调度[J].工业工程,2018,21(02):55-61.
- [64] 张守京,王彦亭.基于改进 NSGA2的柔性车间多目标智能调度问题研究[J].现代制造工程,2020(09):23-31.
- [65] Karaboga D. An idea based on honey bee swarm for numerical optimization, Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey, 2005.
- [66] Sonmez M. Discrete optimum design of truss structures using artificial bee colony algorithm[J]. Structural Multidisciplinary Optimization, 2011. 43(1):85-97.
- [67] Pan Q K. An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling[J]. European Journal of Operational Research, 2016, 250(3):702-714.
- [68] Zhang R, Song S, Wu C. A hybrid artificial bee colony algorithm for the job shop scheduling problem[J]. International Journal of Production Economics, 2013. 141(3):167-178.
- [69] Chang P C, Zhang R, Song S, et al. A multi-objective artificial bee colony algorithm for parallel batch-processing machine scheduling in fabric dyeing processes[J]. Knowledge-Based Systems, 2017. 116(10):114-129.
- [70] Liu M, Lei D. An artificial bee colony with division for distributed unrelated parallel machine scheduling with preventive maintenance[J]. Computers & Industrial Engineering, 2020, 141:106320.

- [71] 毕孝儒,张黎黎,贺拴,等.面向无等待多目标柔性车间调度问题的遗传蜂群优化算法[J].现代计算机(专业版),2015(23):11-16.
- [72] 吴帅,卿兆波.人工蜂群算法求解带有启动时间的批量混合流水车间调度问题[J].煤矿机械,2016,37(11):61-64.
- [73] 郑小操,龚文引.改进人工蜂群算法求解模糊柔性作业车间调度问题[J].控制理论与应用,2020,37(06):1284-1292.
- [74] 李益兵,黄炜星,吴锐.基于改进人工蜂群算法的多目标绿色柔性作业车间调度研究[J].中国机械工程,2020,31(11):1344-1350.
- [75] Allahverdi A, Soroush H M. The significance of reducing setup times setup costs[J]. European Journal of Operational Research, 2008, 187(3):978-984.
- [76] Sharma P, Ajai J. A review on job shop scheduling with setup times[J]. Journal of Engineering Manufacture, 2015, 230(3):517-533.
- [77] Mizuno S, Deeratanasrikul L. Scheduling and lot sizing with sequence-dependent setup: A literature review[J]. Iie Transactions, 2006, 38(1):987-1007.
- [78] Wang S, Wang L. A hybrid estimation of distribution algorithm for the semiconductor final testing scheduling problem[J]. Journal of Intelligent Manufacturing, 2015, 26(5):861-871.
- [79] Akay B B, Karaboga D. Artificial bee colony algorithm variants on constrained optimization[J]. International Journal of Optimization and Control Theories & Applications, 2017, 7(1):98.
- [80] Xue Y, Jiang J, Zhao B, et al. A self-adaptive artificial bee colony algorithm based on global best for global optimization[J]. Soft Computing, 2018, 22(17):2935-2952.
- [81] 何尧, 刘建华, 杨荣华. 人工蜂群算法研究综述[J]. 计算机应用研究, 2018, 319(05):7-12.
- [82] Behnamian J, Ghomi S M T F. Hybrid flowshop scheduling with machine and resource-dependent processing times[J]. Applied Mathematical Modelling, 2011, 35(3):1107-1123.
- [83] Govindan K, Jafarian A, Nourbakhsh V. Bi-objective integrating sustainable order allocation and sustainable supply chain network strategic design with stochastic demand using a novel robust hybrid multi-objective metaheuristic[J]. Computers & Operations Research, 2015, 62(12):112-130.
- [84] Mohammad F F A, Mostafa H K, Seyedali M. Multi-objective stochastic closed-loop supply chain network design with social considerations[J]. Applied Soft Computing, 2018, 71(7):505-525.
- [85] Holsapple C W, Jacob V S, Pakath R. A genetics-based hybrid scheduler for generating static schedules in flexible manufacturing contexts [J]. IEEE Transactions on Systems Man & Cybernetics, 2015, 23(4): 953-972.
- [86] Luo J, Liu Q, Yang Y, et al. An artificial bee colony algorithm for multi-objective optimisation[J]. Applied Soft Computing,2017, 50(11):235-251.
- [87] Xiaoyun Jiang Yi Li. Improved NSGA-II for the Job-Shop Multi-Objective Scheduling Problem [J]. Int J Performability Eng, 2018, 14(5): 891-898.

- [88] 郭书杰, 方兴, 赵鹤群. 算法参数对人工蜂群算法性能的影响[J]. 软件导刊, 2017, 16(4):61-63.
- [89] Y Li, Huang W, Wu R, et al. An improved artificial bee colony algorithm for solving multi-objective low-carbon flexible job shop scheduling problem[J]. Applied Soft Computing, 2020, 95:106544.
- [90] Afzalirad M, Rezaeian J. Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions[J]. Computers & Industrial Engineering, 2016, 98:40-52.
- [91] Gao L, Pan Q K. A shuffled multi-swarm micro-migrating birds optimizer for a multi-resource-constrained flexible job shop scheduling problem[J]. Inform Sciences, 2016,372:655-676.
- [92] Cao Z C, Lin C R, Zhou M C, et al. Scheduling semiconductor testing facility by using cuckoo search algorithm with reinforcement learning and surrogate modeling[J]. IEEE transactions on automation science and engineering, 2019, 16(2):825-837.
- [93] Piroozfard H, Wong K Y, Wong W P. Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm[J]. Resources Conservation and Recycling, 2018, 128:267-283.