

# 带运输时间混合流水车间成组调度的协同进化文化基因算法

袁帅鹏<sup>†</sup>, 李铁克, 王柏琳

(北京科技大学 经济管理学院, 北京 100083; 钢铁生产制造执行系统技术教育部工程研究中心, 北京 100083)

**摘要:** 研究了一类带有序列相关准备时间和阶段间运输时间的混合流水车间成组调度问题, 以最小化最大完工时间为目标建立混合整数线性规划模型, 结合问题特征提出一种协同进化文化基因算法. 算法采用置换序列的方式对工件组间调度、各工件组内工件间调度以及各工件组在各阶段上并行机的指派3个子问题进行统一编码, 基于负载均衡思想和改进的先到先得策略将染色体解码为问题的可行解; 进化过程中采用多种遗传算子执行全域搜索, 并设计了一种基于破坏和重新构造的协同进化局部搜索策略. 通过不同问题规模的数据实验和与对比算法的比较分析, 验证了所提模型和算法的有效性.

**关键词:** 混合流水车间; 成组调度; 运输时间; 协同进化; 文化基因算法

**引用格式:** 袁帅鹏, 李铁克, 王柏琳. 带运输时间混合流水车间成组调度的协同进化文化基因算法. 控制理论与应用, 2023, 40(3): 430–440

DOI: 10.7641/CTA.2022.10779

## Co-evolutionary memetic algorithm for the hybrid flow shop group scheduling with transportation times

YUAN Shuai-peng<sup>†</sup>, LI Tie-ke, WANG Bai-lin

(School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China; Engineering Research Center of MES Technology for Iron & Steel Production, Ministry of Education, Beijing 100083, China)

**Abstract:** This study investigates the hybrid flow shop group scheduling problem with sequence-dependent setup time between groups and job transportation time between machines. A mixed integer linear programming model for minimizing the makespan is established at first. A co-evolutionary memetic algorithm (CMA) is then proposed in view of the problem's feature. In the CMA, the three sub-problems, namely group scheduling, job scheduling within each group and its assignment at each stage, are coded uniformly, and the chromosome is decoded into a feasible solution based on the idea of load balancing and an improved first-come-first-served strategy. A variety of genetic operators are used to perform a global search during the evolution process, and a collaborative optimization local search strategy based on destruction and reconstruction is designed. Experiments with various problem scales are carried out, and the results verify the efficiency and robustness of the proposed algorithm.

**Key words:** hybrid flowshop; group scheduling; transportation time; co-evolutionary; memetic algorithm

**Citation:** YUAN Shuaipeng, LI Tiekke, WANG Bailin. Co-evolutionary memetic algorithm for the hybrid flow shop group scheduling with transportation times. *Control Theory & Applications*, 2023, 40(3): 430–440

## 1 引言

生产调度问题一直是制造科学领域研究的热点, 有效的调度方法和优化技术是提高企业生产效益、实现智能制造的基础和关键. 近年来, 成组调度(group scheduling, GS)得到了国内外学者的大量关注, GS源于单元制造系统, 是一种利用工件在设计和加工上的相似性, 把工件进行分组加工, 以提高生产效率的生

产组织方法. 由于其结构化、高效率 and 低成本的生产优势, GS已经被广泛应用于汽车装配<sup>[1]</sup>、电路板印刷<sup>[2]</sup>和钢铁制造<sup>[3]</sup>等领域. 迄今, 已有的关于GS问题的研究成果主要聚焦于经典的流水车间环境<sup>[4]</sup>, 对混合流水车间环境下GS问题的研究很少. 然而, 相比前者, 混合流水车间环境具有更好的灵活性和稳定性, 能够有效降低瓶颈工序对系统产出的影响, 在现代工

收稿日期: 2021–08–24; 录用日期: 2022–04–29.

<sup>†</sup>通信作者. E-mail: shuaipengyuan@163.com; Tel.: +86 17801002601.

本文责任编辑: 于海斌.

国家自然科学基金项目(71701016, 71231001), 北京市自然科学基金项目(9174038), 中央高校基本科研业务费项目(FRF-BD-20-16A)资助.

Supported by the National Natural Science Foundation of China (71701016, 71231001), the Beijing Natural Science Foundation (9174038) and the Fundamental Research Funds for Central Universities (FRF-BD-20-16A).

业制造系统中扮演着更为重要的角色.因此,对混合流水车间成组调度(hybrid flowshop group scheduling, HFSGS)问题展开研究具有重要的理论意义和实用价值.

与经典调度问题不同,GS需要同时考虑工件组间调度和各工件组内工件间调度两个相关子问题,对于HFSGS问题,还需进一步考虑各工件组在各阶段并行机上的指派过程,是一类复杂的联合决策优化问题. Logendran等<sup>[5]</sup>首次对带有序列不相关准备时间的HFSGS问题展开研究,分析了工件组在某一阶段存在单准备时间和多准备时间两种情况下问题的性质特征,提出一种被称为LN-PT-M的三阶段启发式算法; Zandieh等<sup>[6]</sup>进一步研究了带有序列相关准备时间的HFSGS问题,首先通过启发式规则来确定各工件组在各阶段上的设备指派过程,然后开发了基于遗传和模拟退火的元启发算法分别对组间调度和组内调度两个子问题进行优化求解;针对同样的问题, Keshavarz等<sup>[7]</sup>以最小化总完工时间为目标,构建了问题的混合整数线性规划模型,并提出一种文化基因算法,该算法同样采用分阶段的方式对3个子问题进行独立求解; Ebrahimi等<sup>[8]</sup>则以最小化最大完工时间和总加权拖期为目标,提出两种基于多目标优化的快速非支配排序遗传算法; Feng等<sup>[9]</sup>进一步研究了具有预防性维护的HFSGS问题,从设备和系统两个层面分别构建了问题的数学模型,提出一种嵌套遗传进化的模拟退化算法.

对于考虑阶段间运输时间的HFSGS问题,尚未发现相关研究成果,即使将问题扩展至经典流水车间成组调度领域,也仅有极少数文献可供参考: Liou等<sup>[10]</sup>研究了一类带有序列相关准备时间和阶段间运输时间的流水车间成组调度问题,提出一种混合粒子群和遗传算法,并通过理论分析给出了问题最优解的4种下界;袁帅鹏等<sup>[3]</sup>对带有双向运输时间约束的两阶段流水车间成组调度问题进行了研究,以最大完工时间为目标建立了混合整数线性规划模型,结合问题特征提出一种改进的迭代贪婪算法.

由以上综述可知,目前关于HFSGS问题的研究并不充分,且已有成果主要聚焦于工件组间切换的准备时间,缺乏对阶段间运输过程的深入考虑和分析.然而,在多数实际工业环境下(如钢铁制造、半导体加工等),受到工艺要求的限制和加工环境的制约,工件在阶段间的运输过程是影响生产效率和稳定性的关键因素,致使已有成果在生产管理的实际应用中受到较大局限.因此,非常有必要进一步考虑阶段间运输时间,使得问题更加贴合生产实际,从而指导企业进行精细化管理和调度.算法求解方面,由于HFSGS需要同时考虑3个子问题,属于一类多维度、联合决策优化问题,而已有研究主要采用分阶段的方式对多个子问

题进行单独求解(如文献[5,7]),缺乏对问题特征知识的挖掘以及子问题间关联特征的分析,导致算法的稳健性较差.鉴于以上两点,本文在同时考虑序列相关准备时间和阶段间运输时间的情况下,对HFSGS问题展开研究,首先以最小化Makespan为目标建立问题的数学模型,然后结合问题的特征提出一种协同进化文化基因算法,最后基于不同问题规模的数据实验对算法的有效性进行验证.

## 2 问题描述及数学模型

### 2.1 问题描述

带有序列相关准备时间和阶段运输时间的HFSGS问题可描述如下:1)工件被划分为若干工件组并在多阶段流水车间环境下加工,至少有一个阶段存在多于1台的一致并行机;2)一个工件能且仅能属于一个工件组,对于同工件组内的工件,在每一阶段上只能选取一台并行机进行加工,且加工过程必须连续,不能被其他工件组内的工件打断;3)各阶段上的并行机在切换不同工件组时存在一定的准备时间,该准备时间具有序列相关性,即准备时间同时取决于将要加工的工件组和紧前加工的工件组;4)工件在相邻阶段间的传递过程存在一定的运输时间,运输时间具有工件相关性;5)优化目标为最小化最大完工时间,即Makespan.

该问题需要确定各工件组内工件间的加工顺序、各工件组在各阶段上并行机的指派以及同一并行机上工件组间的加工顺序3个子问题,使得所有工件的最大完工时间尽可能小.若使用FFm表示混合流水车间环境, fmls表示成组调度,  $s_{ijk}$ 表示序列相关准备时间,  $t_{ilk}$ 表示阶段间运输时间,  $C_{\max}$ 表示最大完工时间,根据三元组表示法,本文问题可描述为FFm|fmls,  $s_{ijk}, t_{ilk}$ | $C_{\max}$ .为使问题更加明确,本文基于以下假设:

- 1) 阶段间缓冲区的容量无限;
- 2) 工件在阶段间的运输过程不受运输工具可用性的限制,即无论工件何时完工,均能被实时地运送至下一阶段;
- 3) 生产环境具有稳定性,不考虑设备故障、工件的恶化和学习效应.

### 2.2 数学模型

本节采用0-1整型决策变量的方式建立问题的混合整数线性规划模型,具体如下.

首先,定义以下符号和变量:

1) 索引和集合

$G_i$ 为工件组编号,  $i = 1, 2, \dots, g$ ,  $g$ 为工件组的数量;

$J_{il}$ 为工件组 $G_i$ 内工件的编号,  $l = 1, 2, \dots, n_i$ ,

$n_i$ 为工件组 $G_i$ 内工件的数量;

$k$ 为阶段编号,  $k = 1, 2, \dots, K$ ,  $K$ 为阶段总数;

$M_k$ 为阶段 $k$ 上一致并行机的数量;

2) 问题参数

$p_{ilk}$ 为工件 $J_{il}$ 在阶段 $k$ 上的加工时间;

$s_{i'ik}$ 为阶段 $k$ 上的并行机加工完工件组 $G_{i'}$ 到 $G_i$ 所需要的准备时间;

$t_{ilk}$ 为工件 $J_{il}$ 从阶段 $k$ 到阶段 $k + 1$ 的运输时间,  $k = 1, 2, \dots, K - 1$ ;

$M$ 为足够大的正整数;

3) 决策变量

$x_{i'ik}$ 为0-1决策变量, 在阶段 $k$ 上, 如果工件组 $G_{i'}$ 和 $G_i$ 在同一台并行机上加工, 且 $G_{i'}$ 位于 $G_i$ 紧前的位置, 则为1, 否则为0, 其中 $1 \leq i', i \leq g, i' \neq i$ ; 特别地, 当 $G_i$ 作为首个工件组被加工时表示为 $x_{0ik} = 1$ ;

$y_{il'l}$ 为0-1决策变量, 如果工件 $J_{il}$ 优于工件 $J_{il'}$ 被加工, 则为1; 否则为0, 其中 $1 \leq l' \leq l \leq n_i$ ;

$c_{ilk}$ 为工件 $J_{il}$ 在阶段 $k$ 上的完工时间;

$b_{ilk}$ 为工件 $J_{il}$ 从阶段 $k$ 到阶段 $k + 1$ 的开始运输时间;

$S_{ik}$ 为工件组 $G_i$ 在阶段 $k$ 上的开工时间;

$C_{ik}$ 为工件组 $G_i$ 在阶段 $k$ 上的完工时间;

$C_{\max}$ 为最大完工时间.

混合整数线性规划模型描述如下:

$$\min C_{\max}, \tag{1}$$

$$C_{\max} \geq C_{iK}, \tag{2}$$

$$C_{ik} \geq c_{ilk}, \tag{3}$$

$$\sum_{i'=0, i' \neq i}^g x_{i'ik} = 1, \tag{4}$$

$$\sum_{i'=1, i' \neq i}^g x_{i'ik} \leq 1, \tag{5}$$

$$x_{i'ik} + x_{ii'k} \leq 1, \tag{6}$$

$$\sum_{i=1}^g x_{0ik} = M_k, \tag{7}$$

$$S_{ik} + M(1 - x_{i'ik}) \geq C_{i'k} + s_{i'ik}, \tag{8}$$

$$c_{ilk} - p_{ilk} \geq S_{ik}, \tag{9}$$

$$c_{ilk} - p_{ilk} + M(1 - y_{il'l}) \geq c_{il'k}, \tag{10}$$

$$c_{il'k} - p_{il'k} + M \times y_{il'l} \geq c_{ilk}, \tag{11}$$

$$c_{ilk} - p_{ilk} \geq b_{il(k-1)} + t_{il(k-1)}, \tag{12}$$

$$b_{ilk} \geq c_{ilk}, \tag{13}$$

$$S_{ik}, C_{ik}, c_{ilk}, b_{ilk} \geq 0, \tag{14}$$

$$x_{i'ik}, y_{il'l} \in \{0, 1\}. \tag{15}$$

目标函数(1)表示最小化 $C_{\max}$ ; 约束(2)限制了 $C_{\max}$ 的值域: 不小于所有工件组在最后阶段上的完工

时间; 约束(3)限制了各工件组在各阶段上的完工时间; 约束(4)和约束(6)分别表示在任一阶段, 各工件组均有且仅有一个紧前工件组(包含虚拟工件组 $G_0$ )、至多一个紧后工件组、且两者不能相同; 约束(7)表示在任一阶段的任一并行机上, 有且仅有一个工件组被安排在第一个位置进行加工(紧前工件组为 $G_0$ ); 约束(8)表示如果两个工件组在某阶段上的同一台并行机上加工且存在紧邻关系, 那么后续工件组内任一工件的开工时间不小于紧前工件组的完工时间; 约束(9)表示各工件在各阶段上的开工时间要至少大于所在工件组在该阶段上的开工时间; 约束(10)-(11)为同工件组内工件间的序列约束, 即对于具有先后顺序(不一定紧邻)的两个工件, 后续工件的开工时间至少大于先前工件的完工时间; 约束(12)表示对于任一工件, 只有在当前阶段完工且被运输至下一阶段后才可开始后续阶段的加工; 约束(13)表示各工件的开始运输时间至少大于其完工时间; 约束(14)-(15)为决策变量的取值范围. 该模型已通过CPLEX数学规划软件验证了其正确性.

为更加清楚地阐述该调度问题, 给定如下实例: 工件组数量为3, 各工件组内工件的数量均为2, 阶段数量为2, 两阶段并行机的数量分别为2, 1, 组间切换的准备时间、各工件组内工件的加工时间和运输时间分别如表1( $G_0$ 为虚拟工件组)和表2所示, 并假定设备指派和工件组间调度序列方案为 $M_{11} : G_1 - G_2$ ;  $M_{12} : G_3$ ;  $M_{21} : G_1 - G_2 - G_3$ , 组内工件间调度序列均为 $J_{i1} - J_{i2}(i = 1, 2, 3)$ , 图1给出了该问题实例的调度甘特图.

表1 组间切换准备时间

	$k = 1$			$k = 2$		
	$G_1$	$G_2$	$G_3$	$G_1$	$G_2$	$G_3$
$G_0$	2	2	3	2	4	2
$G_1$	-	3	5	-	3	4
$G_2$	5	-	3	2	-	2
$G_3$	2	4	-	2	3	-

表2 工件加工时间和阶段间运输时间

		$G_1$			$G_2$			$G_3$		
		$J_{11}$	$J_{12}$	$J_{21}$	$J_{22}$	$J_{31}$	$J_{32}$			
加工时间	$k = 1$	2	3	1	3	5	4			
	$k = 2$	3	2	3	2	2	3			
运输时间	$k = 1$	2	2	2	4	3	5			

3 协同进化文化基因算法

当所有工件的运输时间均为0时, 问题FFm|fmls,

$s_{ijk}, t_{ilk} | C_{max}$  很容易归结为具有强NP难特性的混合流水车间成组调度问题<sup>[6]</sup>, 因此  $FFm|fmls, s_{ijk}, t_{ilk} | C_{max}$  同样具有强NP难特性且更为复杂. 精确算法

虽在理论上可以得到最优解, 但难以求解大规模问题, 无法满足工业应用的需要, 而基于智能优化的元启发式算法为快速求解该类问题提供了一种有效途径.

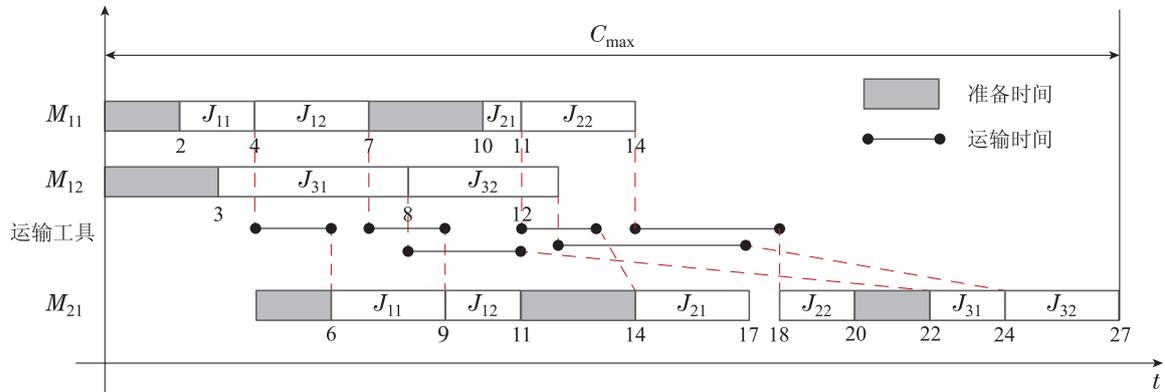


图1 给定实例调度甘特图

Fig. 1 Scheduling Gantt-chart for given instance

文化基因算法(memetic algorithm, MA)是一种新型的、基于全局和局部搜索相结合的元启发算法, 其设计思路与经典遗传算法类似, 不同的是MA摒弃了遗传算法中具有盲目性的随机变异操作, 在全局搜索的基础上加入了基于问题特征的局部搜索策略, 使每次迭代的全部个体都能够达到局部最优<sup>[11]</sup>. MA已被证实在很多领域要优于经典遗传算法, 且已在调度类的组合优化问题方面取得了较好的效果<sup>[12-14]</sup>. 据此, 本文结合MA的优化机制和  $FFm|fmls, s_{ijk}, t_{ilk} | C_{max}$  问题的特征, 提出一种改进的协同进化文化基因算法(co-evolutionary memetic algorithm, CMA). CMA采用置换序列的方式对3个子问题进行统一编码, 基于负载均衡思想和改进的先到先得策略将染色体解码为问题的可行解; 进化过程中引入多种遗传算子来进行全域搜索, 设计了一种基于破坏和重新构造的协同进化局部搜索策略, 并给出一种重置机制以避免陷入局部最优, 具体实施过程如下.

### 3.1 编码解码策略

针对HFSGS问题, 一个完整的编码方案需要包括各工件组在各阶段上并行机的指派、同一阶段各并行机上工件组间的加工顺序和各工件组内工件间的加工顺序3部分. CMA将染色体编码为由  $1+g$  个置换序列向量组成的整数序列, 序列总长度为  $g + \sum_{i=1}^g n_i$ . 其中第一部分(记为  $X$ )用来确定各工件组在第一阶段上并行机的选择以及同一并行机上工件组间的加工顺序; 后  $g$  部分(记为  $Y_i (i = 1, \dots, g)$ )用来确定各工件组内工件间的加工顺序. 假定工件组数量  $g = 5$ , 各工件组内工件的数量依次为3, 3, 4, 4, 2, 图2给出了的一个染色体的具体编码方案.

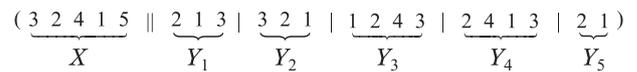


图2 染色体编码方案

Fig. 2 Chromosome coding scheme

为了保证种群的多样性, CMA采用随机方式生成初始种群, 即对于每一条染色体, 首先随机生成一个长度为  $g$  的置换序列向量, 得到子集  $X$ ; 然后依次随机生成  $g$  个长度为  $n_i$  的置换序列向量, 得到子集  $Y_i$ , 随后将  $X$  和  $Y_i$  整合得到一条完整的染色体. 对于每一条染色体, 解码时首先基于负载均衡思想确定第一阶段各并行机加工工件组的数量(记为  $\sigma_h (h = 1, 2, \dots, M_1)$ ), 然后结合子集  $X$  确定各工件组在第一阶段上并行机的选择及同一并行机上工件组间的加工顺序, 再由子集  $Y_i$  确定各工件组内工件间的加工顺序; 对于后续阶段, 根据上一阶段的完工顺序, 采用改进的先到先得策略确定各工件组在各阶段上的并行机的选择. 具体实施步骤如下:

**步骤1** 使用式(16)计算  $\sigma_h$  的值, 其中  $\lfloor \cdot \rfloor$  和  $\lceil \cdot \rceil$  分别为向下和向上取整操作:

$$\sigma_h = \begin{cases} \lceil g/M_1 \rceil, & 1 \leq h \leq \text{mod}(g, M_1), \\ \lfloor g/M_1 \rfloor, & \text{mod}(g, M_1) < h \leq M_1. \end{cases} \quad (16)$$

**步骤2** 根据  $\sigma_h$  的值构造辅助向量

$$GV = [\underbrace{1, 1, \dots, 1}_{\sigma_1}, \underbrace{2, 2, \dots, 2}_{\sigma_2}, \dots, \underbrace{M_1, M_1, \dots, M_1}_{\sigma_{M_1}}].$$

**步骤3** 由式(17)确定各工件组在第一阶段所选取并行机的编号, 记为  $I_1(i)$ , 即

$$I_1(i) = GV(X(i)). \quad (17)$$

**步骤4** 对于在同一并行机上加工的工件组集, 按照子集  $X$  中出现的顺序依次加工.

**步骤5** 对于后续阶段 $k(k = 2, \dots, K)$ , 使用式(18)确定工件组 $G_i$ 所选取的并行机编号 $I_k(i)$ , 并按照上一阶段完工的顺序依次加工, 其中 $\delta_h$ 表示当前调度序列中并行机 $h$ 上最后加工的工件组( $G_i$ 的紧前工件组).

$$I_k(i) = \arg \min_{1 \leq h \leq M_k} \{C_{\delta_h k} + s_{\delta_h i k}\}. \quad (18)$$

由上述实施步骤可以看出, 该解码策略充分考虑了并行机的负载均衡, 并在传统先到的策略基础上进一步考虑了工件组在各并行机上准备时间的差异性, 使得算法具有较大的概率得到更优解.

### 3.2 全局优化策略

MA的全局优化策略可以从遗传算法、模拟退火算法、粒子群算法等多种现成的进化算法中选择, 其中遗传算法作为一类经典的智能优化算法, 能够根据问题特征和编码策略灵活设置进化算子, 具有较强的全局搜索能力. 因此针对本文问题, 作者借鉴遗传算法的进化思想来执行全局优化操作, 具体如下.

#### 3.2.1 选择策略

选择操作是驱动算法向优势区域进化的关键过程. 为提升CMA的全局搜索能力, 算法采用改进的精英保留策略执行选择操作, 即在每一次迭代中, 首先将当前全局最优个体直接复制给下一代, 然后选取该全局最优个体和种群中的其他每一个个体作为父代进行后续交叉变异操作, 并将产生的优势个体存入下一代种群中.

#### 3.2.2 交叉、变异策略

交叉的目的是将父代染色体中的优势基因结构传递给子代, 进而获取更优解. 对于置换序列的编码方式, 常用的交叉策略包括两点交叉(two-point crossover, TPX)、部分映射交叉(partially mapped crossover, PMX)和顺序交叉(order-based crossover, OBX)等, 其中OBX算子已被证实在求解流水线调度问题时要优于其他算子<sup>[15-16]</sup>, 通过预备实验发现, 该算子同样适用于本文问题且具有较好的求解效果. 因此, 本文采用OBX算子对每条染色体中执行交叉操作, 具体思路如下: 对于父代个体 $P_1$ 和 $P_2$ , 首先随机选取若干基因位, 将 $P_1$ 中对应的基因直接放入子代个体, 再将 $P_2$ 中未使用的基因依次插入 $C_1$ 的空闲位置, 进而得到第一个子代个体 $C_1$ ; 然后交换 $P_1$ 和 $P_2$ 的位置, 重复上述操作得到子代个体 $C_2$ . 图3给出了OBX算子的实施过程.

变异操作的目的是通过不确定性的扰动来增加种群多样性, 当前常用于置换序列编码的变异算子有插入、交换和反转3种. 通过预备实验发现3种变异操作对算法性能的影响并不大, 因此在每次迭代过程中, 从3种变异算子中随机选取其一执行变异操作, 3种变

异算子的执行过程如图4所示.

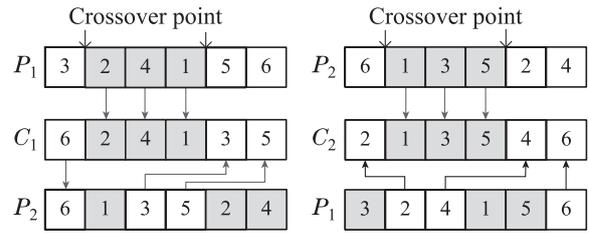


图3 OBX算子执行过程

Fig. 3 Process of OBX operator

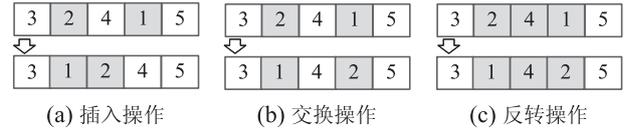


图4 3种变异算子执行过程

Fig. 4 Process of three mutation operators

鉴于子集 $X$ 和 $Y_i$ 分别代表不同的子问题, 在每一次进化过程中, CMA采用如下策略来生成完整的子代个体: 首先对父代个体中的子集 $Y_i$ 逐一进行交叉变异操作, 并将父代中的子集 $Y_i$ 和新生成的子集 $\text{new}_Y_i$ 进行比较, 选取较优者进入子代个体; 待所有 $Y_i$ 交叉变异完成后, 以新生成的子集 $\text{new}_Y_i$ 为基础, 对子集 $X$ 执行交叉变异操作, 进而得到完整的子代个体, 算法1给出了交叉变异的执行过程. 整个过程的算法复杂度为 $O(\sum_{i=1}^g n_i)$ , 需要对新个体进行评价的次数为 $4 \times (g + 1)$ 次.

#### 算法1 CMA交叉变异执行过程

输入: 父代个体 $(X, Y)$ 和 $(\bar{X}, \bar{Y})$ ;

输出: 子代个体 $(\text{new}_X, \text{new}_Y)$ 和 $(\text{new}_\bar{X}, \text{new}_\bar{Y})$ ;

for  $i = 1 : g$  do

$\pi \leftarrow Y_i$ ;

$\bar{\pi} \leftarrow \bar{Y}_i$ ;

$\pi_1, \pi_2 \leftarrow$  对个体 $\pi$ 和 $\bar{\pi}$ 执行交叉操作;

$\pi'_1, \pi'_2 \leftarrow$  对个体 $\pi_1$ 和 $\pi_2$ 执行变异操作;

$\text{new}_\pi, \text{new}_{\bar{\pi}} \leftarrow$  从 $(\pi_1, \pi_2, \pi'_1, \pi'_2)$ 中选取最优的两个个体;

$\text{new}_Y_i \leftarrow \text{new}_\pi$ ;

$\text{new}_{\bar{Y}_i} \leftarrow \text{new}_{\bar{\pi}}$ ;

end

$X_1, X_2 \leftarrow$  对个体 $X$ 和 $\bar{X}$ 执行交叉操作;

$X'_1, X'_2 \leftarrow$  对个体 $X_1$ 和 $X_2$ 执行变异操作;

$\text{new}_X \leftarrow \arg \min_X (\{C_{\max}(X_1, \text{new}_Y), C_{\max}(X'_1, \text{new}_Y)\})$ ;

$\text{new}_{\bar{X}} \leftarrow \arg \min_{\bar{X}} (\{C_{\max}(X_2, \text{new}_{\bar{Y}}), C_{\max}(X'_2, \text{new}_{\bar{Y}})\})$ ;

输出:  $(\text{new}_X, \text{new}_Y)$ 和 $(\text{new}_{\bar{X}}, \text{new}_{\bar{Y}})$ .

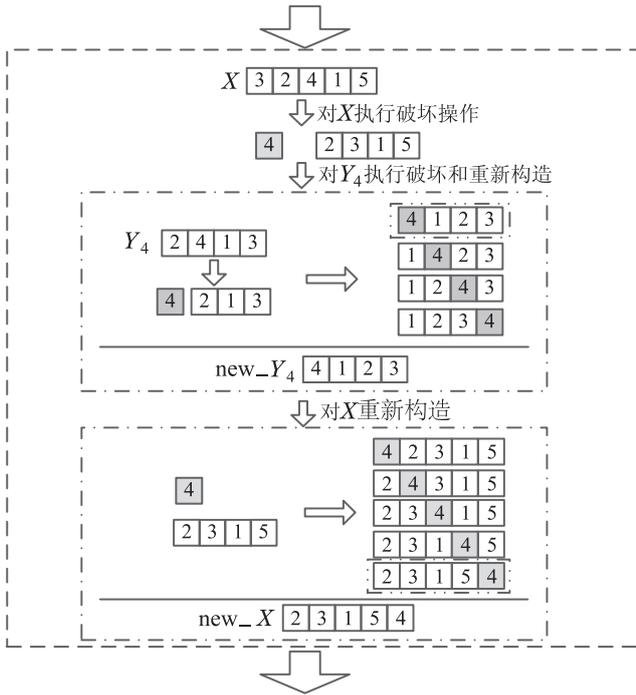
### 3.3 协同进化局部搜索策略

局部搜索是进一步挖掘优势解的关键途径. 由编码策略可知, 尽管子集 $X$ 和 $Y$ 分别代表不同的子问题,

但两者具有很强的关联性, 为提高算法搜索到优质解的概率, 本文借鉴迭代贪婪算法的进化思想<sup>[17]</sup>, 设计了一种基于破坏和重新构造的协同进化局部搜索策略对当前种群内的每一条染色体进行深度搜索. 具体思路如下: 首先对当前解的子集 $X$ 执行规模为1的破坏操作, 即从子集 $X$ 中随机选取并移除一个基因; 然后对该基因对应的工件序列 $Y_i$ 执行破坏和重新构造, 并从新得到的 $n_i$ 个工件序列中选取最优者替换当前工件序列; 随后, 以更新后的子集 $Y$ 为基准, 将移出的基因重新插入子集 $X$ 所有可能的位置, 并选取所得到工件组序列中的最优者替换当前子集 $X$ , 进而得到完整的新解.

图5给出了所设计协同进化局部搜索策略的具体执行过程, 根据其执行过程可以看出, 在每次进化过程中, 该协同进化局部搜索策略能够针对特定的工件组及其工件序列进行联合优化. 另外, 由图5可知, 该局部搜索策略的算法复杂度为 $O(g + \max_{v_i}(n_i))$ .

当前解: ( 3 2 4 1 5 || 2 1 3 | 1 3 2 | 1 2 4 3 | 2 4 1 3 | 1 2 )  
 $X \quad Y_1 \quad Y_2 \quad Y_3 \quad Y_4 \quad Y_5$



新解: ( 2 3 1 5 4 || 2 1 3 | 1 2 3 | 1 2 4 3 | 4 1 2 3 | 1 2 )  
 $X \quad Y_1 \quad Y_2 \quad Y_3 \quad Y_4 \quad Y_5$

图5 协同进化局部搜索策略执行过程

Fig. 5 Process of co-evolutionary local search strategy

### 3.4 重置机制

为避免算法陷入局部最优并指导算法探索新的优势区域, 本文采用如下重置机制重新构造当前种群: 1) 保留前20%的优势个体不变; 2) 使用随机方式生成其他80%的个体. 该重置机制的触发条件是当前最优解在规定代数(记为 $T$ )内无法由更优解替代. 本文在

后续实验部分分析了 $T$ 取不同值时算法的性能, 进而确定 $T$ 的最佳取值.

### 3.5 算法流程

基于以上算法设计, CMA 求解问题  $FFm|fmls, s_{ijk}, t_{ilk}|C_{max}$  的实施步骤如下:

**步骤 1** 初始化算法参数, 包括种群规模 $p\_size$ , 交叉概率 $p_c$ , 变异概率 $p_m$ , 最优解保持代数 $T$ .

**步骤 2** 结合第3.1节的编码策略随机生成初始种群.

**步骤 3** 采用第3.2节的选择、交叉、变异策略对当前种群进行全局优化操作.

**步骤 4** 使用第3.3节的基于破坏和重新构造的协同进化局部搜索策略对当前种群进行深度搜索.

**步骤 5** 判断是否达到重置限制, 若达到则启用第3.4节的重置机制重新构造当前种群.

**步骤 6** 判断算法的运行时间是否达到最大运行时间限制, 若未达到, 转至步骤3; 否则, 输出最优解及目标值.

## 4 实验结果与分析

### 4.1 实验设计

实验首先通过多因素方差分析确定CMA算法中的关键参数(包括进化策略和参数值), 然后基于小问题规模测试算例将CMA的运行结果与混合整数线性规划模型取得的最优解进行对比; 最后从文献中选取了3种与本文问题相关且具有较好效果的元启发式算法, 并基于大问题规模的测试算例将CMA的运行结果与3种算法进行对比. 实验中所有算法均使用MATLAB R2015a编程实现, 模型采用ILOG CPLEX 12.7.0 数学规划软件进行求解, 算法测试环境为 Intel i5-6200U/CPU 2.40GHz/8.0 GB.

由于未发现可直接应用于本文问题的基准数据, 以文献[7](研究带有序列相关准备时间HFGSGS问题)中的实验数据作为参考, 并对其进行修正和扩充, 得到以下实验数据: 阶段数量 $K \in \{2, 3, 4, 5, 6\}$ , 工件组数量 $g \in \{3, 4, 5, 10, 15, 20\}$ , 各工件组内工件的数量 $n_i \in \{3, 4, 5, 7, 10\}$ . 根据阶段数量、工件规模(所有工件组内工件的总数)将以上数据分为9组小问题规模的实验( $K \in \{2\}$ ,  $g \in \{3, 4, 5\}$ 和 $n_i \in \{3, 4, 5\}$ 的正交组合)和24组大问题规模的实验( $K \in \{3, 4, 5, 6\}$ ,  $g \in \{10, 15, 20\}$ 和 $n_i \in \{7, 10\}$ 的正交组合). 实验中其他问题参数设置如下: 每一阶段并行机的数量 $M_k \in DU[1, 3]$ , 工件加工时间 $p_{ilk} \in DU[5, 75]$ , 序列相关准备时间 $s_{i'ik} \in DU[5, 50]$ , 阶段间运输时间 $t_{ilk} \in DU[5, 50]$ , 数据格式 $DU[a, b]$ 表示介于 $a$ 和 $b$ 之间的离散均匀分布. 实验测试算例可通过以下地址访问得到<https://>

github.com/17801002601/TestData.

实验采用相对偏差率(percentage relative difference, PRD)<sup>[18]</sup>来衡量算法性能, PRD计算公式为

$$PRD(A) = \frac{C_{\max}(A) - \text{ref}}{\text{ref}} \times 100\%, \quad (19)$$

其中:  $C_{\max}(A)$ 表示当前测试算例算法A得到的 $C_{\max}$ 值, ref为当前测试算例的参考值, 计算方法如下: 对于小问题规模的算例, 以模型取得的最优值作为参考值; 对于大问题规模的测试算例, 以实验中所有算法得到的最好解作为参考值. 同一测试算例对于各算法的参考值是唯一(相等)的, 因此对于每组测试算例, PRD值越小, 说明该算法越接近于最好解.

### 4.2 策略有效性检验及参数设置

CMA中的关键策略包括交叉算子、变异算子和局部搜索策略, 算法参数包括种群规模 $p\_size$ 、交叉概率 $p_c$ 、变异概率 $p_m$ 和最优解保持代数 $T$ . 针对交叉算子, 设置3个对比策略: TPX, PMX和OBX; 变异算子同样设置3个对比策略: 插入操作(记为Insert), 交换操作(记为Exchange)和反转操作(记为Reverse); 局部搜索设置2个对比策略: 采用和不采用所设计局部优化策略. 通过预备实验, 种群规模 $p\_size$ , 交叉概率 $p_c$ , 变异概率 $p_m$ 和最优解保持代数 $T$ , 4个参数均设置3个水平值, 具体如表3所示. 综上, 会产生 $3 \times 3 \times 2 \times 3 \times 3 \times 3 \times 3 = 1458$ 种具有不同参数配置的CMA算法.

实验使用多因素方差分析 (analysis of variance, ANOVA)<sup>[19]</sup>来验证CMA中所设计关键策略的有效性

并确定参数值. 对于24组大问题规模实验, 每组随机产生1个测试算例, 使用所有CMA算法分别求解, 将所有算例每种算法的PRD均值作为响应变量, 采用 $F$ 值和 $P$ 值分析实验结果, 其中 $F$ 值用来衡量该因素对结果的影响程度,  $P$ 值用来检验各因素不同水平值是否存在显著性差异. ANOVA分析结果如表4所示, 各因素95%置信区间见图6-7.

表3 算法参数水平值设置

Table 3 Combinations of parameter values

参数	水平值		
	1	2	3
$p\_size$	10	20	40
$p_c$	0.5	0.7	0.9
$p_m$	0.05	0.1	0.3
$T$	5	10	15

表4 ANOVA分析结果

Table 4 Results of ANOVA

因素	总平方和	自由度	均方	$F$ 值	$P$ 值
交叉算子	10.21	2	5.10	18.19	1.51E-06
变异算子	0.08	2	0.04	0.56	0.57
局部搜索	8.04	1	4.02	91.20	6.06E-14
$p\_size$	74.05	2	37.03	31.58	3.63E-06
$p_c$	24.57	2	12.29	11.06	8.16E-05
$p_m$	2.50	2	1.25	7.30	0.002
$T$	10.52	2	5.27	25.55	8.07E-09

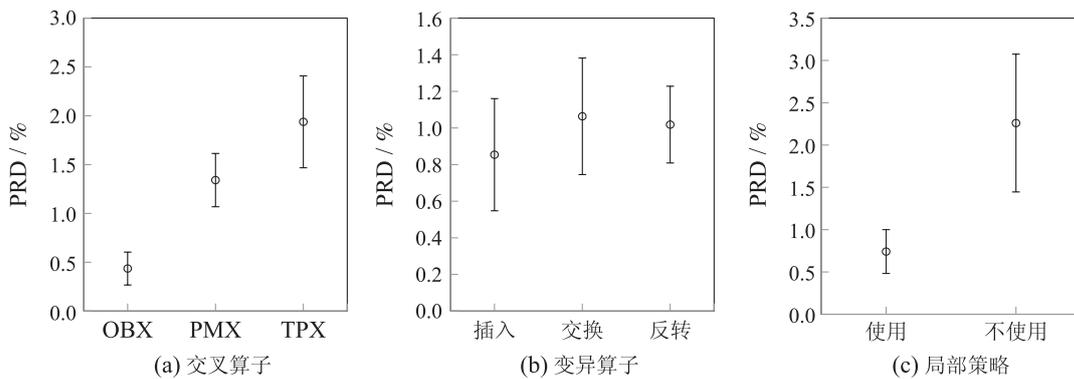


图6 CMA中关键策略各因素95%置信区间

Fig. 6 Mean with 95% confidence intervals of key strategies in CMA

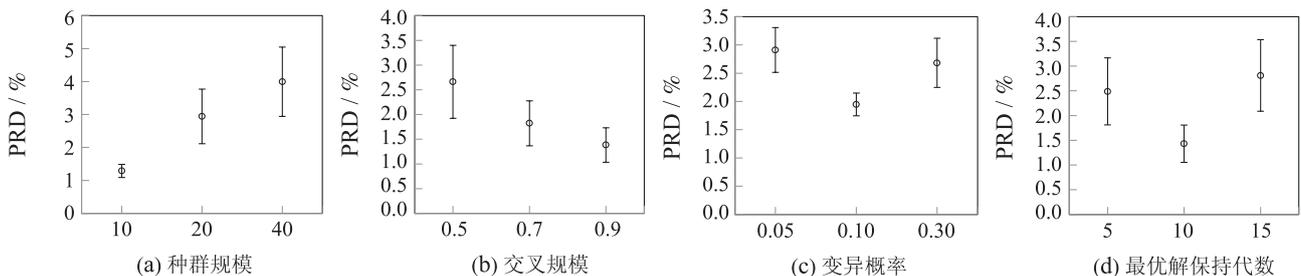


图7 CMA中算法参数各因素95%置信区间

Fig. 7 Mean with 95% confidence intervals of parameters in CMA

由表4可以看出, 3种策略中局部搜索策略具有最大的 $F$ 值, 这表明该策略对算法性能影响最大, 图6(c)给出了使用和不使用局部搜索策略CMA算法PRD均值的95%置信区间, 显然使用局部搜索策略具有更低的PRD均值, 这证实了该策略的有效性; 交叉算子获得的 $F$ 值排名第2, 结合图6(a)可以看出, 与其他两种交叉算子相比, OBX的PRD均值最低, 这说明该算子的求解性能要优于其他两种交叉算子; 对于变异算子, 由表4可以看出其 $F$ 值仅为0.56, 且 $P$ 值大于0.05, 这说明3种变异算子不存在显著差异性, 因此, 在CMA中采用随机方式从3种变异算子中选取.

算法参数方面, 种群规模 $p\_size$ 获得了最大的 $F$ 值, 表明该参数对算法性能影响最大, 随后依次是最大维持代数 $T$ 、交叉概率 $p_c$ 和变异概率 $p_m$ , 进一步结合图7可以直观看出: 当 $p\_size = 10, p_c = 0.9, p_m = 0.1$ 和 $T = 10$ 时CMA算法具有最好的求解性能, 因此以下实验采用此参数设置.

### 4.3 最优性检验

本节使用9组小问题规模实验来检验所提模型的正确性, 并观察CMA相对于模型取得最优值的偏差情况. 每组实验随机生成10个测试算例, 对于每个测试算例, 分别使用CPLEX和CMA进行求解, 设定CMA的运行时间为 $100 \times K \times \sum_{i=1}^g n_i$  ms, 设定CPLEX的运行时间为600 s. 采用PRD和最优解算例数两个指标来检验CMA的性能, 实验结果如表5所示.

表 5 最优性测试结果  
Table 5 Optimality test results

K	g	$n_i$	均值( $C_{max}$ )		PRD/%	最优解算例数
			CPLEX	CMA		
2	3	3	<b>299.7</b>	<b>299.7</b>	0	10/10
2	3	4	<b>363.2</b>	363.9	0.193	9/10
2	3	5	<b>415.6</b>	416.8	0.289	8/10
2	4	3	<b>352.3</b>	355.7	0.965	9/10
2	4	4	<b>454.4</b>	455.4	0.22	9/10
2	4	5	<b>481.1</b>	481.6	0.104	9/10
2	5	3	<b>399.7</b>	402.1	0.6	6/10
2	5	4	<b>476.9</b>	481.1	0.881	9/10
2	5	5	<b>579</b>	583.3	0.561	2/4
平均			424.66	426.62	0.424	-
合计			-	-	-	71/84

由表5可以看出, 对于第一组实验, 10组测试算例CMA均得到了与CPLEX一样的 $C_{max}$ 值(PRD均值为0), 这证实了本文所设计模型的正确性, 同时也证实了CMA在求解小问题规模测试算例方面的有效性. 整体上, 9组实验CMA获得最大PRD均值为0.881%, 总体PRD均值仅为0.424%, 这表明对于小问题规模测试

算例CMA得到的值与最优解的间隙非常的小. 另一方面, 由最优解算例数可以看出, 84组测试算例中共71组得到了最优解, 这进一步表明对于大多数测试算例(约84.5%)CMA均能在有限的时间内得到最优解. 随着问题规模的增大, 最后1组实验中有6组测试算例CPLEX已无法在规定的时间内得到模型的最优解, 这是因为问题本身的强NP难特性和模型的复杂性导致的, 这从侧面也证实了所设计算法相较于精确模型的优越性. 由以上结果可以得出结论: 本文提出的CMA算法在求解小问题规模测试算例时具有很好的优化性能.

### 4.4 与其他元启发式算法对比

为测试CMA的性能, 本文选取了两种针对HFGS问题提出的具有较好效果的元启发式算法, 分别为文献[7]中的MA和文献[9]中的嵌套遗传进化模拟退火算法(simulated annealing embedded genetic algorithm, SAGA), 并从最新文献中选取了两种针对成组调度问题提出的主流元启发算法, 分别为文献[3]中的迭代贪婪算法(iterated greedy algorithm, IGA)和文献[20]中的多点模拟退火算法(multi-start simulated annealing, MSA), 将这4种算法应用于本文问题, 4种对比算法均采用原文推荐的参数设置. 需要说明的是, 由于本文问题与文献中所研究问题存在一定的差别, 实验过程中对3种算法的评价函数做了适应性调整.

针对24组大问题规模实验, 每组随机生成10组测试算例, 对于每组测试算例, 设置5种算法的运行时间均为 $100 \times K \times \sum_{i=1}^g n_i$  ms, 并使用每种算法单独运行10次, 计算各组实验下所有算例5种算法的PRD均值和标准差, 以及每组实验CMA获得最好解(参考值)的次数, 结果如表6所示. 此外, 为研究CMA与其他3种算法的求解质量在统计意义上的差异性, 采用配对 $t$ 检验(paired t-test,  $\alpha = 0.05$ )对每组实验两两算法PRD均值的差异性进行显著性检验, 检验结果见表7.

由表6的统计结果可以看出, 24组测试实验中CMA得到的PRD均值全部低于其他4种算法, 且所有实验CMA的总体PRD均值仅为1.12%, 明显低于MA(2.36%), SAGA(2.14%), IGA(1.80%)和MSA(2.58%), 这表明CMA的求解质量要优于其他4种算法. 对于PRD值的标准差, 仅有1组实验( $K = 4, g = 10, n_i = 7$ )CMA略高于算法IGA(CMA位于5种算法中的第2位, 与第1位IGA算法的差值为0.07%), 其他实验CMA均为5种算法中的最低者, 且其总体标准差为0.73%, 同样低于MA(1.43%), SAGA(1.32%), IGA(1.05%)和MSA(1.06%), 说明CMA在获取较优解的同时, 求解性能也相对稳定. 此外, 与文献[7]中提出的MA算法相比, 本文设计的CMA算法在求解质量和稳定性方面都得到了提升, 这得益于CMA中提出的基于破坏和重

新构造的协同进化局部搜索策略,该策略能够在全局优化的基础上,进一步探索较好解的邻域区域,此举提高了算法搜索到更优解的概率;与文献[3]提出的IGA算法相比,CMA算法的优势在于IGA算法仅使用一个初始解进行全局和局部的搜索,算法的搜索的效率依赖于初始解的优劣,且较容易陷入局部最优状态,

而CMA算法则通过种群进化的机制来优化搜索,同时通过重置机制避免算法陷入局部最优状态,因此在求解质量方面也有较大提升.另外,根据表7的检验结果可知,24组实验CMA与其他4种算法通过配对 $t$ 检验得到的 $p$ 值均小于0.05,这进一步说明与其他4种算法相比,CMA的求解质量存在显著的优越性.

表6 5种算法PRD均值、标准差统计结果

Table 6 Results of five algorithms in PRD mean and standard deviation

$K$	$g$	$n_i$	CMA		MA		SAGA		IGA		MSA	
			平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
3	10	7	<b>2.35</b>	<b>0.85</b>	5.51	1.13	6.98	0.93	5.85	0.89	2.99	1.05
3	10	10	<b>0.98</b>	<b>0.71</b>	2.6	1.45	2.05	0.72	1.45	1.04	2.87	1.02
3	15	7	<b>1.07</b>	<b>0.57</b>	2.29	1.69	1.83	1.99	1.71	0.83	2.83	0.91
3	15	10	<b>0.9</b>	<b>0.67</b>	2.15	1.16	1.88	1.27	1.82	1.19	2.71	0.98
3	20	7	<b>1.28</b>	<b>0.75</b>	2.04	1.62	2.09	1.72	1.61	1.34	2.79	1.17
3	20	10	<b>1.16</b>	<b>0.61</b>	2.03	1.51	1.73	1.09	1.65	1.46	2.67	1.06
4	10	7	<b>1.08</b>	0.49	2.05	0.92	1.57	0.86	1.29	<b>0.41</b>	2.64	0.83
4	10	10	<b>1.05</b>	<b>0.83</b>	3.52	1.35	2.78	0.91	1.99	0.96	2.52	1.12
4	15	7	<b>0.71</b>	<b>0.44</b>	1.51	1.84	1.5	1.07	1.29	0.95	2.61	0.68
4	15	10	<b>0.65</b>	<b>0.24</b>	1.03	1.28	0.96	0.87	0.83	0.48	2.49	0.77
4	20	7	<b>1.05</b>	<b>0.88</b>	3.06	1.86	2.43	1.72	1.9	0.95	2.51	1.1
4	20	10	<b>1.31</b>	<b>1.35</b>	2.01	1.93	1.85	1.75	1.66	1.42	2.37	1.84
5	10	7	<b>0.98</b>	<b>0.69</b>	2.99	1.58	2.76	1.54	2.27	0.97	2.78	1.22
5	10	10	<b>1.09</b>	<b>0.44</b>	1.51	0.54	1.26	0.73	1.15	0.65	2.64	0.79
5	15	7	<b>1.21</b>	<b>0.46</b>	1.94	1.53	1.56	1.23	1.41	1.01	2.42	0.73
5	15	10	<b>0.95</b>	<b>0.57</b>	1.71	0.96	1.58	0.87	1.29	0.63	2.66	0.61
5	20	7	<b>0.66</b>	<b>0.72</b>	2.42	1.27	2.06	1.02	1.94	0.96	2.3	0.89
5	20	10	<b>0.94</b>	<b>0.65</b>	1.98	1.71	1.86	1.05	1.13	0.88	1.9	0.93
6	10	7	<b>1.38</b>	<b>1.26</b>	2.33	1.72	2.09	1.75	1.81	1.72	2.93	1.47
6	10	10	<b>1.5</b>	<b>1.33</b>	2.76	1.38	2.48	1.56	2.43	1.49	2.75	1.6
6	15	7	<b>1.23</b>	<b>0.69</b>	2.56	1.41	2.36	2.11	1.69	1.2	2.71	1.15
6	15	10	<b>0.85</b>	<b>0.77</b>	2.05	0.98	1.95	1.45	1.37	1.07	2.78	1.23
6	20	7	<b>1.35</b>	<b>0.72</b>	2.29	1.93	1.79	1.75	1.67	1.64	2.23	0.92
6	20	10	<b>1.18</b>	<b>0.81</b>	2.23	1.57	2.04	1.73	1.91	0.96	1.92	1.26
		均值	<b>1.12</b>	<b>0.73</b>	2.36	1.43	2.14	1.32	1.8	1.05	2.58	1.06

为测试不同问题规模对算法性能的影响,根据表6的实验结果进一步汇总了不同阶段和工件规模(所有工件组内工件的总数,即 $\sum_{i=1}^g n_i$ )下5种算法的总体PRD均值,结果如表8所示.由其结果可以看出:当阶段数量相同时,随着工件规模的增加,5种算法的总体PRD均值呈现逐渐递减的趋势(如:当 $K=3$ 时,CMA算法在3种工件规模下的总体PRD均值分别为1.67%,1.18%和1.03%),这表明对于本文所研究问题,5种算法的求解质量均具有较好的稳健性,其中CMA始终低于其他4种算法,这表明CMA在不同工件规模下均能得到相对较好的解;当工件规模相同时,随着

阶段数量的增加,5种算法的总体PRD值整体呈现先减后增的趋势(如对于小工件规模的测试算例,4个阶段下CMA的总体PRD值分别为1.67%,1.07%,1.04%和1.44%),这说明阶段数量会对算法性能造成一定的影响,但对于任意的阶段和工件规模,CMA在5种算法中始终处于较低的水平,这进一步证实了CMA在求解质量和稳定性方面的优越性.

综上所述,可以得出结论:针对问题FFm|fmls, $s_{ijk}, t_{ilk}|C_{\max}$ ,本文提出的CMA算法在求解质量和稳定性方面要优于文献中提出的MA, SAGA, IGA和MSA.

表 7 5种算法PRD均值的配对t检验结果

Table 7 Paired t-test results of PRD means for five algorithms

K	g	n <sub>i</sub>	CMA & MA		CMA & SAGA		CMA & IGA		CMA & MSA	
			p-value	p < 0.05	p-value	p < 0.05	p-value	p < 0.05	p-value	p < 0.05
3	10	7	0.013	是	0.038	是	0.012	是	0.009	是
3	10	10	0.021	是	0.011	是	0.015	是	0.007	是
3	15	7	0.032	是	0.037	是	0.005	是	<0.001	是
3	15	10	<0.001	是	0.003	是	0.008	是	0.012	是
3	20	7	<0.001	是	0.027	是	0.014	是	0.006	是
3	20	10	0.015	是	0.002	是	0.021	是	0.011	是
4	10	7	0.027	是	0.012	是	<0.001	是	<0.001	是
4	10	10	0.009	是	0.020	是	0.001	是	<0.001	是
4	15	7	0.001	是	0.032	是	0.004	是	0.015	是
4	15	10	0.004	是	0.018	是	0.006	是	0.003	是
4	20	7	0.010	是	0.004	是	0.013	是	0.001	是
4	20	10	0.025	是	0.002	是	<0.001	是	0.020	是
5	10	7	<0.001	是	<0.001	是	0.021	是	0.006	是
5	10	10	0.001	是	<0.001	是	0.032	是	0.006	是
5	15	7	0.001	是	0.007	是	0.003	是	0.004	是
5	15	10	0.009	是	0.009	是	0.027	是	0.010	是
5	20	7	<0.001	是	<0.001	是	0.001	是	<0.001	是
5	20	10	<0.001	是	0.018	是	<0.001	是	0.003	是
6	10	7	<0.001	是	0.029	是	0.016	是	0.019	是
6	10	10	<0.001	是	0.014	是	<0.001	是	0.012	是
6	15	7	0.011	是	0.042	是	0.008	是	0.008	是
6	15	10	0.008	是	0.017	是	0.01	是	0.005	是
6	20	7	0.020	是	0.026	是	0.002	是	0.014	是
6	20	10	0.006	是	0.019	是	0.231	是	0.009	是

表 8 不同阶段和工件规模下5种算法总体PRD均值对比

Table 8 Overall PRD mean values of five algorithms at different stages and job scales

K	问题规模	CMA	MA	SAGA	IGA	MSA
3	小	1.67	4.06	4.52	3.65	2.93
	中	1.18	2.17	1.96	1.66	2.81
	大	1.03	2.09	1.81	1.74	2.69
4	小	1.07	2.79	2.18	1.64	2.58
	中	0.88	2.29	1.97	1.6	2.56
	大	0.98	1.52	1.41	1.25	2.43
5	小	1.04	2.25	2.01	1.71	2.71
	中	0.94	2.18	1.81	1.68	2.36
	大	0.95	1.85	1.72	1.21	2.28
6	小	1.44	2.55	2.29	2.12	2.84
	中	1.29	2.43	2.08	1.68	2.47
	大	1.02	2.14	2	1.64	2.35

小:  $\sum_{i=1}^g n_i \leq 100$ ; 中:  $100 < \sum_{i=1}^g n_i \leq 140$ ; 大:  $\sum_{i=1}^g n_i > 140$ .

### 5 结论

本文研究了一类带有序列相关准备时间和阶段间运输时间的混合流水车间成组调度问题, 该问题具有广泛的工业应用背景, 但尚未发现相关研究成果. 为

求解此调度问题, 以最小化最大完工时间为目标构建了混合整数线性规划模型, 鉴于问题的强NP难特性和联合决策特征, 提出一种改进的协同进化文化基因算法. 算法采用置换序列的方式对3个子问题进行统一

编码, 基于负载均衡思想和改进的先到先得策略将染色体解码为问题的可行解; 进化过程中通过多种遗传算子来进行全局搜索, 设计了一种基于破坏和重新构造的协同进化局部搜索策略, 并提出一种重置机制以避免算法陷入局部最优状态. 实验阶段, 首先通过多因素方差分析确定了算法中的关键策略的有效性和最佳参数值, 然后基于小问题规模测试算例验证了所提算法的最优性(与最优解的偏差情况), 最后基于大问题规模测试算例证实了所提算法的高效性和稳健性. 与已有文献相比, 本文研究成果能够进一步丰富成组调度理论和方法体系, 在实践上能够为钢铁生产, 汽车装配等具有成组加工特征制造企业的调度管理提供技术支持, 算法中设计的协同进化局部搜索策略也能够为其他具有联合决策特征生产调度问题的求解提供新的思路和方法.

本文仅考虑了运输工具的单向运输时间, 即假定工件在阶段间运输过程不受运输工具可用性的限制, 这并不完全符合实际工业生产过程, 下一步将进一步考虑运输工具数量和运输能力存在限制时的混合流水线车间成组调度问题, 结合问题特征构建调度规则, 并以此为基础设计更为高效的求解算法.

## 参考文献:

- [1] SALMASI N, LOGENDRAN R, SKANDARI M R. Total flow time minimization in a flow shop sequence-dependent group scheduling problem. *Computers & Operations Research*, 2010, 37(1): 199 – 212.
- [2] CELANO G, COSTA A, FICHERA S. Constrained scheduling of the inspection activities on semiconductor wafers grouped in families with sequence-dependent set-up times. *International Journal of Advanced Manufacturing Technology*, 2010, 46(5): 695 – 705.
- [3] YUAN Shuaipeng, LI Tieke, WANG Bailin, et al. Co-evolutionary iterated greedy algorithm for the two-stage flow shop group scheduling problem. *Systems Engineering – Theory & Practice*, 2020, 40(10): 2707 – 2716.  
(袁帅鹏, 李铁克, 王柏琳, 等. 两阶段流水线车间成组调度问题的迭代贪婪算法. *系统工程理论与实践*, 2020, 40(10): 2707 – 2716.)
- [4] NEUFELD J S, GUPTA J, BUSCHER U. A comprehensive review of flowshop group scheduling literature. *Computers & Operations Research*, 2016, 70(1): 56 – 74.
- [5] LOGENDRAN R, CARSON S, HANSON E. Group scheduling in flexible flow shops. *International Journal of Production Economics*, 2005, 96(2): 143 – 155.
- [6] ZANDIEH M, DORRI B, KHAMSEH A R. Robust metaheuristics for group scheduling with sequence-dependent setup times in hybrid flexible flow shops. *The International Journal of Advanced Manufacturing Technology*, 2009, 43(7): 767 – 778.
- [7] KESHAVARZ T, SALMASI N, VARMAZYAR M. Minimizing total completion time in the flexible flowshop sequence – dependent group scheduling problem. *Annals of Operations Research*, 2015, 226(1): 351 – 377.
- [8] EBRAHIMI M, GHOMI S, KARIMI B. Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. *Applied Mathematical Modelling*, 2014, 38(9): 2490 – 2504.
- [9] FENG H, XI L, XIAO L, et al. Imperfect preventive maintenance optimization for flexible flowshop manufacturing cells considering sequence-dependent group scheduling. *Reliability Engineering & System Safety*, 2018, 176(8): 218 – 229.
- [10] LIOU C, HSIEH Y. A hybrid algorithm for the multi-stage flow shop group scheduling with sequence dependent setup and transportation times. *International Journal of Production Economics*, 2015, 170(12): 258 – 267.
- [11] CHEN Xiang, ZHU Chuanjun, ZHANG Chaoyong. An open shop scheduling problem based on memetic algorithm. *Industrial Engineering Journal*, 2018, 21(6): 16 – 22.  
(陈祥, 朱传军, 张超勇. 基于文化基因算法的开放车间调度问题研究. *工业工程*, 2018, 21(6): 16 – 22.)
- [12] HAJINEJAD D, SALMASI N, MOKHTARI R. A fast hybrid particle swarm optimization algorithm for flowshop sequence dependent group scheduling problem. *Scientia Iranica*, 2011, 18(3): 759 – 764.
- [13] TANG Qiuhua, ZHENG Peng, ZHANG Liping, et al. Multi-buffered steelmaking production scheduling with heuristic rules and memetic algorithm. *Computer Integrated Manufacturing Systems*, 2015, 21(11): 2955 – 2963.  
(唐秋华, 郑鹏, 张利平, 等. 融合启发式规则和文化基因算法的多缓冲炼钢—连铸生产调度. *计算机集成制造系统*, 2015, 21(11): 2955 – 2963.)
- [14] LUAN Zhiwei, LI Tieke, WANG Bailin. Learnable memetic algorithm for hot rolling batch scheduling of seamless steel tube. *Computer Integrated Manufacturing Systems*, 2019, 25(11): 2753 – 2761.  
(栾治伟, 李铁克, 王柏琳. 无缝钢管热轧批量调度问题的学习型文化基因算法. *计算机集成制造系统*, 2019, 25(11): 2753 – 2761.)
- [15] YU C, SEMERARO Q, MATTA A. A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility. *Computers & Operations Research*, 2018, 100(1): 211 – 229.
- [16] YAURIMA V, BURTSEVA L, TCHERNYKH A. Hybrid flowshop with unrelated machines, sequence – dependent setup time, availability constraints and limited buffers. *Computers & Industrial Engineering*, 2009, 56(4): 1452 – 1463.
- [17] WANG Ling, PAN Zixiao. Scheduling optimization for flow-shop based on deep reinforcement learning and iterative greedy method. *Control and Decision*, 2021, 36(11): 2609 – 2617.  
(王凌, 潘子肖. 基于深度强化学习与迭代贪婪的流水线调度优化. *控制与决策*, 2021, 36(11): 2609 – 2617.)
- [18] WANG B, HUANG K, LI T. Two-stage hybrid flowshop scheduling with simultaneous processing machines. *Journal of Scheduling*, 2018, 21(4): 387 – 411.
- [19] KHARE A, AGRAWAL S. Scheduling hybrid flowshop with sequence-dependent setup times and due windows to minimize total weighted earliness and tardiness. *Computers & Industrial Engineering*, 2019, 135(9): 780 – 792.
- [20] CHENG C Y, POURHEJAZY P, YING K C, et al. New benchmark algorithms for no-wait flowshop group scheduling problem with sequence-dependent setup times. *Applied Soft Computing*, 2021, 111: 107705.

## 作者简介:

袁帅鹏 讲师, 目前研究方向为生产计划与调度、智能优化, E-mail: shuaipengyuan@163.com;

李铁克 教授, 目前研究方向为智能制造、先进制造管理, E-mail: tieke@ustb.edu.cn;

王柏琳 副教授, 目前研究方向为生产计划与调度、高级计划排程、运筹及智能优化, E-mail: wangbl@ustb.edu.cn.