

改进人工蜂群算法求解分布式柔性作业车间调度问题

吴 锐, 郭顺生, 李益兵[†], 王 磊, 许文祥

(武汉理工大学 机电工程学院, 武汉 430070)

摘 要: 针对分布式柔性作业车间调度问题的特点, 提出一种改进人工蜂群算法. 首先, 建立以最小化最大完工时间为优化目标的分布式柔性作业车间调度优化模型; 然后, 改进基本人工蜂群算法以使其适用于求解分布式柔性作业车间调度问题, 具体的改进包括设计一种包含三维向量的编码方案, 结合问题特点针对性地设计多种策略用于种群初始化, 在雇佣蜂改良搜索操作中设计多种有效的进化操作算子, 并在跟随蜂搜索操作中引入基于关键路径的局部搜索算子以提升算法的局部搜索能力; 最后, 利用扩展柔性作业车间通用测试集得到的测试数据设计实验验证算法性能, 使用正交试验法优化算法参数设置. 仿真实验结果表明, 改进后的人工蜂群算法能有效求解分布式柔性作业车间调度问题.

关键词: 分布式柔性作业车间; 加工单元分配; 工件排序; 人工蜂群算法; 关键路径; 最大完工时间
中图分类号: TP18 **文献标志码:** A

Improved artificial bee colony algorithm for distributed and flexible job-shop scheduling problem

WU Rui, GUO Shun-sheng, LI Yi-bing[†], WANG Lei, XU Wen-xiang

(School of Mechanical and Electrical Engineering, Wuhan University of Technology, Wuhan 430070, Cina)

Abstract: This paper analyzes the characteristics of the distributed and flexible job-shop scheduling problem and proposes an improved artificial bee colony algorithm for solving the problem. Firstly, a scheduling model is established to minimize the makespan. Then, some improvements are applied to the basic artificial bee colony algorithm so that it can solve this problem effectively, including a three-dimension encoding scheme, effective population initialization method based on the characteristics of the distributed and flexible job-shop scheduling problem, many evolutionary operators are designed for the employed bee search phase, and what's more, in onlooker bee phase, a local search operator based on the critical path is introduced to improve the local search capability of the algorithm. Finally, an experiment is designed to verify the performance of the algorithm based on the test data expanded from common benchmark of the flexible job-shop scheduling problem, and orthogonal test is used to optimize the parameters in the proposed algorithm. The results show that the improved artificial bee colony algorithm can effectively solve the problem.

Keywords: distributed and flexible job-shop scheduling; manufacturing unit allocation; job sequencing; artificial bee colony algorithm; critical path; makespan

0 引 言

生产调度是企业优化资源配置的重要手段之一, 是合理利用企业资源和最大限度满足客户需求的关键, 因此生产调度优化问题一直是研究的热点. 近年来, 由于市场竞争日益激烈, 产品需求频繁多变, 需要企业具备快速响应市场的能力, 众多企业纷纷引进柔性制造系统, 逐渐具备了柔性制造的能力^[1]. 基于此背景的柔性作业车间调度问题(Flexible job shop scheduling problem, FJSP) 一经提出就成为了研究热

点, 众多学者对其展开了系列研究并取得了众多成果^[2-5]. 与此同时, 在当今全球化制造的趋势下, 制造业正从传统的集中式制造模式向分布式制造模式转变, 产品的制造往往不再由一个企业或生产基地全部完成, 而是交由位于不同地域的多个企业或生产基地共同来完成. 分布式制造可以充分利用多个企业或工厂的资源, 通过资源的合理配置、优化组合及共享, 以合理的成本快速实现产品的生产制造^[6]. 由于分布式制造模式应用越来越广泛, 而且针对单车间的柔性

收稿日期: 2018-03-21; 修回日期: 2018-07-03.

基金项目: 湖北省科技支撑计划项目(2015BAA063); 中央高校基本科研业务费专项基金项目(2016-YB-020, 2016III024).

责任编辑: 王凌

[†]通讯作者. E-mail: ahlyb@whut.edu.cn.

作业车间调度机制、方法和策略难以在分布式制造模式中取得理想的效果,结合并扩展分布式制造模式与FJSP的分布式柔性作业车间调度问题(Distributed and flexible job shop scheduling problem, DFJSP)被提出并且逐渐被学者关注并研究.与FJSP不同的是,在DFJSP中,一个独立的制造工厂或生产基地被视为一个生产单元,本文将其称为柔性加工单元(Flexible manufacturing unit, FMU),单个的FMU即可视作一个柔性作业车间,多个FMU需要互相协作来完成给定的生产任务,同时各个FMU具有不同数量及不同加工能力的机器,导致其加工容量及加工效率互不相同.DFJSP比FJSP更为复杂,因为每个FMU内的工序排序及机器选择问题就相当于一个FJSP,在此基础上还需解决生产任务在FMU之间的分配问题,显然DFJSP属于NP-hard问题,研究其优化求解方法具有重要的学术意义.在工程上,DFJSP比FJSP更加贴近现实的生产状况,研究成果也更能指导当前的企业生产,因此寻求针对DFJSP的高效调度方法和调度机制具有重要的应用价值.

目前,国内外学者对DFJSP的研究相对较少,并且该问题求解复杂度高,传统的数学规划方法无法在有限的时间内找到问题的最优解,因此现有的研究都集中在利用智能优化算法求解DFJSP上.较早对DFJSP进行研究的有Chan等^[7-8],研究了利用主导基因的遗传算法解决DFJSP;Giovanni等^[9]也采用了遗传算法求解DFJSP,并提出了4种变异算子以改善算法性能;黄英杰等^[10]将粒子群算法与目标级联法混合,基于此提出了层次化的DFJSP优化模型;杨江波等^[11]结合车间实际生产情况,提出了一种基于目标级联法和遗传算法的DFJSP层次调度模型;Ziaee等^[12]针对DFJSP问题特性,为机器限制的时间、工件平均耗时和工序加工耗时等因素赋予权重,并平衡了这3个因素对DFJSP问题的影响,使算法能快速锁定较好解所在的区域;Lu等^[13]针对DFJSP中制造单元分配、机器选择和工序排序3个子问题,将问题归纳为在三维空间中寻找最优解的问题.

综上所述,目前对DFJSP求解方法的研究较少,且当前研究主要都是用遗传算法解决DFJSP问题,但遗传算法有易陷入早熟收敛和收敛慢的缺点.此外,在现有研究中较少针对DFJSP问题特点来设计进化操作算子,大多是利用一定的映射方法将DFJSP这一离散优化问题转换为连续优化问题,这一做法虽然可以简化算法中进化操作算子的操作,但是并没有有效利用问题的信息,导致搜索操作并没有针对性和方向性.目前,国内外学者都在不断寻求新的智能优

化算法和有效地搜索操作来更有效地求解DFJSP.人工蜂群算法(Artificial bee colony algorithm, ABC)是Karaboga^[14]于2005年提出的一种群体智能优化算法,其基本思想是模拟自然界蜂群采蜜的智能行为.由于其具有结构简单、参数较少、易于实现等特点,受到了众多学者的关注和研究^[15-17],也广泛地应用于生产调度领域^[18-21],而且ABC已被证明是求解FJSP的一种有效方法^[22-25].目前尚未见到将ABC用于求解DFJSP的研究,本文利用ABC求解DFJSP,并根据问题特点针对性地改进ABC以使其能高效求解DFJSP.

1 分布式柔性作业车间调度问题

1.1 符号定义

问题描述及模型定义中所使用的符号含义如下:
索引号.

i : 工件索引号, $i \in \{1, 2, \dots, N\}$;

j : 工序索引号;

k : 加工设备索引号;

r : 加工顺序索引号;

l : FMU索引号, $l \in \{1, 2, \dots, Q\}$.

变量.

J_i : 第 i 个待加工工件;

O_{ij} : 工件 i 中第 j 道工序;

U_l : 第 l 个FMU;

M_k : 第 k 个加工设备;

N : 工件总数目;

p_i : 工件 i 中工序总数目;

Q : FMU总数目;

m_l : U_l 中加工设备数目;

$t_{i,j}^{l,k}$: U_l 中第 k 台设备加工工件 i 中第 j 道工序所需加工时间;

L : 一个足够大的正数.

决策变量.

$x_{i,j}^{l,k}$: 若工件 i 的第 j 道工序在 U_l 中第 k 台设备上加工,则等于1,否则等于0;

$z_{i,j}^{l,k,r}$: 若工件 i 的第 j 道工序在 U_l 中第 k 台加工设备上的加工顺序索引号为 r ,则等于1,否则等于0;

$S_{i,j}$: 工件 i 中第 j 道工序的起始加工时间;

$C_{i,j}$: 工件 i 中第 j 道工序的结束加工时间;

C_i : 工件 i 的完工时间;

C_{\max} : 全局最大完工时间;

$E_{l,k,r}$: U_l 中第 k 台设备上加工顺序索引号为 r 的工件的起始加工时间;

$q_{l,k}$: 由 U_l 中第 k 台设备加工的工序总数目.

1.2 问题描述

本文研究的DFJSP问题可被描述如下:待加工工件集合 $J = \{J_1, J_2, \dots, J_N\}$ 需要被安排在FMU集合 $U = \{U_1, U_2, \dots, U_Q\}$ 中加工;第 l 个FMU中装备有多台加工设备 $M = \{M_1, M_2, \dots, M_{m_l}\}$,每个待加工工件 J_i 包含 p_i 道工序,工序之间遵循工艺约束,每道工序均可在一台或多台加工设备上加工;一个工件只能被安排在同一FMU中加工,不能将一个工件内的工序分散到不同FMU中加工.由此可看出,DFJSP包含3个子问题:为工件选择FMU,确定工序加工顺序,为工序选择加工设备.图1为DFJSP的示意图.

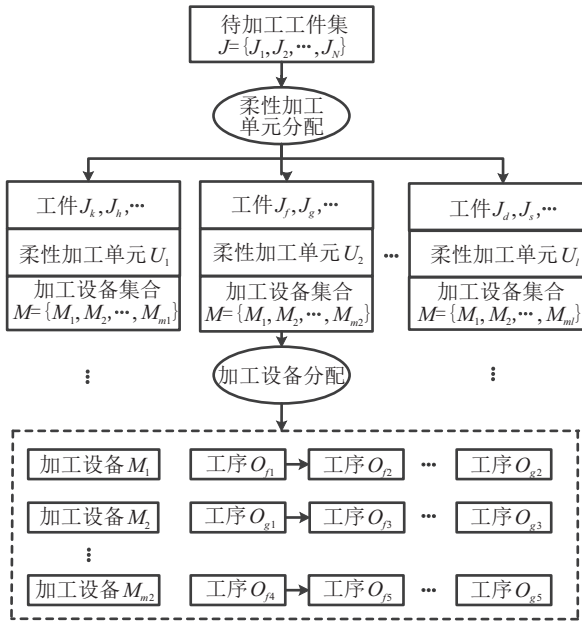


图1 分布式柔性制造车间系统示意图

表1为一个具有3个待加工工件和两个FMU的例子,表1中的数值代表对应工序的所需加工时间.例如,表1中第2行描述的是工件 J_1 中的工序 O_{12} 在各台设备上的加工时间,该行中第4列的数值为5,即表示由 U_1 中的加工设备 M_2 加工工序 O_{12} 上需消耗5个单位时间;“-”表示该加工设备不具备加工该道工序的能力.

表1 3个工件和2个FMU的DFJSP示例

工件	工序	U_1			U_2		
		M_1	M_2	M_3	M_1	M_2	M_3
J_1	O_{11}	2	1	3	3	-	2
	O_{12}	3	5	-	3	3	3
	O_{13}	3	5	2	2	1	-
J_2	O_{21}	4	6	2	5	4	5
	O_{22}	3	2	7	5	4	3
J_3	O_{31}	3	1	4	3	6	4
	O_{32}	-	3	4	5	3	4
	O_{33}	4	4	2	-	3	4

1.3 模型建立

大多数企业在生产调度环节中最希望快捷地完成生产任务,因此本文设定最小化最大完工时间(makespan)为优化目标,并以此建立调度优化模型.本文建立的模型是基于如下假设.

假设1 在0时刻所有FMU中的所有加工设备均可用,所有工件均可被加工.

假设2 工件之间没有加工顺序约束,同一工件的工序有加工顺序约束.

假设3 基于运输成本等考虑,同一工件中的工序只选择同一个FMU来加工.

假设4 一道工序只能选择一台加工设备来加工,且只能被加工一次.

假设5 一台加工设备在同一时刻只能加工一道工序.

假设6 不考虑工序加工中断的情况.

假设7 不考虑运输时间、准备时间、释放时间等.

基于上述假设,以最小化makespan为优化目标的调度优化模型建立如下:

$$\min C_{\max} = \max_{i=1}^N \{C_i\}. \tag{1}$$

$$C_i = C_{i,p_i}; \tag{2}$$

$$C_{i,j} = S_{i,j} + \sum_{l=1}^Q \sum_{k=1}^{m_l} x_{i,j}^{l,k} \times t_{i,j}^{l,k}, j \in \{1, 2, \dots, p_i\}; \tag{3}$$

$$S_{i,j+1} \geq C_{i,j}, j \in \{1, 2, \dots, p_i - 1\}; \tag{4}$$

$$E_{l,k,r+1} \geq E_{l,k,r} + \sum_{i=1}^N \sum_{j=1}^{p_i} t_{i,j}^{l,k} \times z_{i,j}^{l,k,r},$$

$$l \in \{1, 2, \dots, Q\}, k \in \{1, 2, \dots, m_l\},$$

$$r \in \{1, 2, \dots, q_{l,k} - 1\}; \tag{5}$$

$$\sum_{j=1}^{p_i} \sum_{k=1}^{m_l} x_{i,j}^{l,k} \in \{0, p_i\}; \tag{6}$$

$$\sum_{l=1}^Q \sum_{k=1}^{m_l} x_{i,j}^{l,k} = 1, j \in \{1, 2, \dots, p_i\}; \tag{7}$$

$$\sum_{r=1}^{q_{l,k}} z_{i,j}^{l,k,r} = x_{i,j}^{l,k}, j \in \{1, 2, \dots, p_i\},$$

$$k \in \{1, 2, \dots, m_l\}; \tag{8}$$

$$S_{i,j} \geq 0, E_{l,k,r} \geq 0; \tag{9}$$

$$x_{i,j}^{l,k} \in \{0, 1\}, z_{i,j}^{l,k,r} \in \{0, 1\}. \tag{10}$$

其中:式(1)是目标函数,表示最大完工时间是所有工件完工时间的最大值;等式约束(2)表示工件的完工时间是最后一道工序的完工时间;式(3)表示工序的

完工时间等于开始加工时间加上实际加工时间;不等式约束(4)是工序约束,同一工件内的工序必须按顺序加工;式(5)是加工设备约束,同一台加工设备同一时间只能加工一道工序;式(6)表示一个工件的所有工序只能在同一个FMU内加工;式(7)表明一道工序只能选择一个FMU里的一台加工设备来加工;式(8)限定一道工序只能被所选择的加工设备加工一次;式(9)和(10)表示决策变量的取值范围。

2 改进人工蜂群算法求解DFJSP

2.1 基本人工蜂群算法概述

在基本ABC算法中,人工蜂群包括引领蜂、跟随蜂和侦查蜂3种类型的蜜蜂。假设在D维空间中,种群规模为SN(SN/2 = 引领蜂个数 = 跟随蜂个数 = 蜜源数目),第i个蜜源记为 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ 。每个蜜源代表优化问题的一个候选解,花蜜的数量反映了解的质量^[26]。人工蜂群搜索最优蜜源的过程如下:1)引领蜂对当前蜜源进行改良搜索,产生新蜜源,贪婪选择较优蜜源;2)跟随蜂根据引领蜂分享的信息选择一个蜜源,进行邻域搜索,贪婪选择较优蜜源;3)引领蜂放弃蜜源,转变成侦查蜂,并随机搜索产生新的蜜源。

搜索过程中,跟随蜂根据引领蜂分享的信息,以轮盘赌的方式按下式选择一个蜜源:

$$P_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \quad (11)$$

其中: fit_i 是蜜源的适应度值,可按下式计算得出:

$$fit_i = \begin{cases} 1/(1 + f_i), & f_i \geq 0; \\ 1 + abs(f_i), & f_i < 0. \end{cases} \quad (12)$$

这里 f_i 是第i个蜜源的目标函数值。

引领蜂和跟随蜂按下式搜索新蜜源:

$$v_{ij} = x_{ij} + r(x_{ij} - x_{kj}) \quad (13)$$

其中:k是随机产生的整数, $k \in \{1, 2, \dots, SN\}$ 且 $k \neq i, j \in \{1, 2, \dots, D\}; r \in [-1, 1]$ 是一个随机数。若在达到一定尝试次数(limit)后,蜜源质量没有得到提高,则引领蜂放弃该蜜源,转变为侦查蜂,按下式随机产生新蜜源:

$$x_{ij} = l_j + rand(0, 1)(u_j - l_j) \quad (14)$$

其中 l_j 和 u_j 分别是变量 x_{ij} 的下界和上界。

2.2 改进人工蜂群算法整体框架

基本ABC算法只适用于求解连续型问题,但DFJSP是典型的离散组合问题,故需要做离散化的改进才能使其适用于求解DFJSP,同时由于DFJSP具有

复杂性,单一操作很难有效地求解该问题。因此,本文提出改进人工蜂群算法(Improved artificial bee colony algorithm, IABC),设计混合初始化策略、多种交叉与变异组成的侦查蜂改良搜索操作以及基于关键路径的局部搜索操作,从而使算法的探索能力和局部搜索能力得到有效的结合和平衡。所提出的IABC整体流程如图2所示。

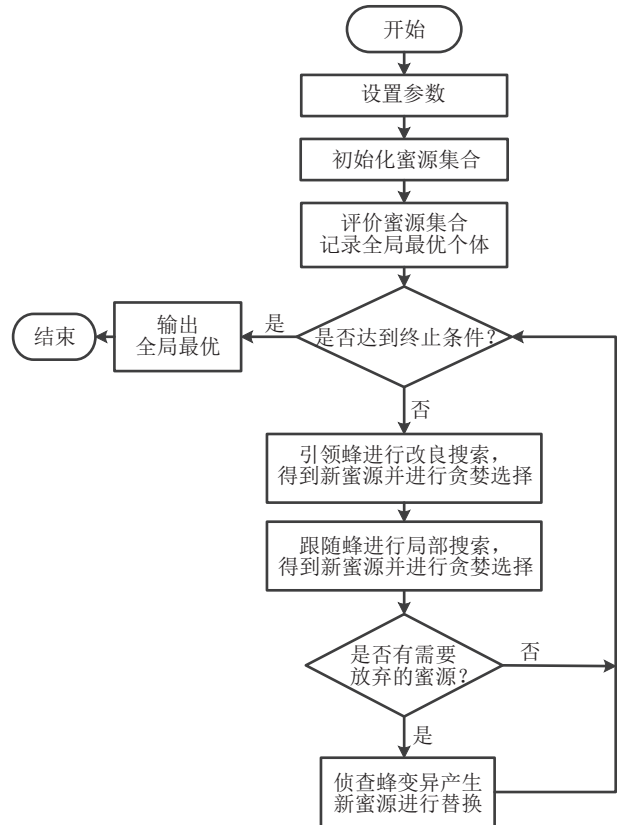


图2 IABC整体流程图

2.3 算法具体设计

2.3.1 编码和解码

由DFJSP的问题特性可知,DFJSP的编码由3部分组成,分别是工序排序向量(Operation sequence vector, OV)、FMU选择向量(FMU selection vector, UV)及加工设备选择向量(Machine assignment vector, MV)。3个向量的长度均等于所有工件中的工序数之和,对应表1中的示例,其一个可行解编码如图3所示。

	O_{21}	O_{11}	O_{31}	O_{32}	O_{12}	O_{22}	O_{13}	O_{33}
OV	2	1	3	3	1	2	1	3
	Job 1		Job 2		Job 3			
UV	2	2	2	1	1	2	2	2
	Job 1		Job 2		Job 3			
	O_{11}	O_{12}	O_{13}	O_{21}	O_{22}	O_{31}	O_{32}	O_{33}
MV	1	3	2	2	3	1	1	2

图3 编码示意

OV中每个工件的工序都由该工件的工件索引

号表示,某个工件索引号第几次出现就代表该工件中的第几道工序,如图3中OV的第5个元素为第2次出现的1,即代表工件1中的第2道工序 O_{12} ;FMU选择向量由FMU索引号组成,其对应的是将工件按索引号由小到大从左到右排列时各工件所选的FMU,需要确保一个工件只能选择一个FMU,即实现数学模型中的约束(6),如图3所示,UV中前3个元素值为2,表示工件1中的3个工序在 U_2 中加工,紧接着的2个元素值为1,即工件2中的2个工序在 U_1 中加工,余下的同理;MV的编码原理与UV类似,可看作是存在一条虚拟工序排序向量V,该向量内工序优先按工件索引号,再按工序索引号从小到大排列,MV中元素值代表的是V中对应位置工序所选择的设备,如图3中MV的第5个元素代表工件2中的第2道工序安排在第3台加工设备上加工,需要注意所选设备的合法性。

从可行解编码中并不能直接得到调度方案,需要通过解码操作将可行解编码转换为调度方案.针对上述编码方案设计的解码操作主要思路为将可行解编码以FMU为单位分解为多个FJSP解码子问题,然后针对各子问题分别解码.具体实施步骤如下。

Step 1: 以各个FMU为单位,在OV、MV中提取对应的工序号和加工设备号,期间保持元素先后顺序不变.由于编码中确保了每个工件只选择一个FMU,在以FMU为单位将整体编码方案分解为多个子编码时就可确保得到的所有子编码都是确定的.以图3所示的编码示例为例, U_2 加工工件1和3,因此提取OV中工件1和3内所有工序组成工序加工子向量,提取出MV中所有对应加工设备得到设备分配子向量,构成 U_2 的子编码;同理提取出 U_1 对应的子向量,结果如图4所示。

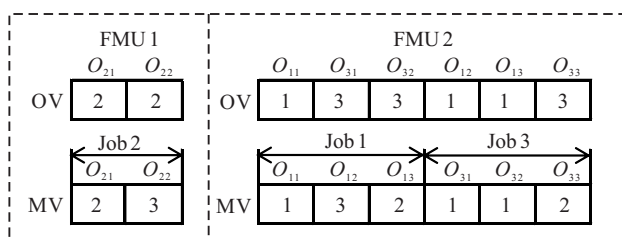


图4 分解后的子编码示意

Step 2: 针对各解码子问题分别进行解码,利用FJSP中成熟的解码方法进行^[22].该解码方法中的“左移策略”就是使各工序尽可能早地开始加工,这种在安排工序时都尽量往“左”凑的方法能确保解码过程中每道工序的排布具有唯一性,由此可知子编码的解码结果也具有唯一性.各解码子问题解码完成后可得到各FMU内的调度方案,综合在一起就可得到整个DFJSP的调度方案,同时可得到各FMU内的本地

最大完工时间,取其中的最大值即为优化目标全局最大完工时间。

基于上述的解码方法,对图3中的编码方案进行解码操作,解码完成后的调度方案甘特图如图5所示。

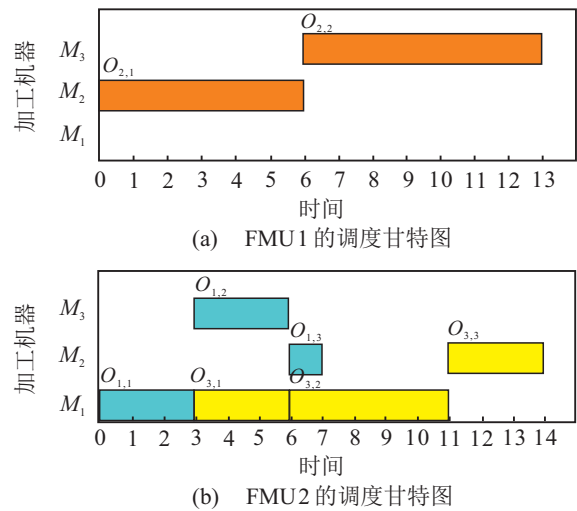


图5 解码完成后的调度方案甘特图

2.3.2 种群初始化方法

初始解的质量对人工蜂群算法求解速度和求解质量有非常大的影响.为提高初始解的质量及其质量的稳定性,本文基于DFJSP的特点提出若干策略用于种群的初始化操作。

在初始化种群时,首先,基于如下策略生成所有蜜源中的OV,各生成50%: 1) 工序随机排序; 2) 优先选择剩余工序多的工件,若存在多选,则从待选内随机选择.然后,生成UV,使用以下策略分别生成种群规模的40%、40%及20%: 1) 优先选择加工设备总数多的FMU,若存在多选,则比较各FMU内所有可加工设备所需的加工时间平均值,优先选择平均时间少的FMU,仍存在多选时从待选内随机选择; 2) 优先选择已分配工件数目少的FMU,若存在多选,则比较各已分配工件内工序总数,优先选择工序总数少的FMU,仍存在多选时从待选内随机选择; 3) 随机选择一个FMU.生成MV的策略思路与上述基本一致,也分别生成种群规模的40%、40%及20%: 1) 优先选择已分配工序数目少的加工设备,若存在多选,则从待选内随机选择; 2) 优先选择加工时间短的加工设备,若存在多选,则从待选内随机选择; 3) 随机选择一个加工设备。

2.3.3 引领蜂搜索操作

在引领蜂对蜜源进行改良搜索操作中,针对OV、UV及MV各自的特点分别设计了有效的交叉和变异操作,以实现高效的搜索操作。

父代选择操作:进行交叉的两个父代其中一个为当前蜜源X,另外一个父代采用锦标赛选择方法来

选择,即从蜜源集合中随机选择3个蜜源(不包括当前蜜源 X),从被选的3个蜜源中选择最优的一个作为待交叉父代 X_{best} ,被选择过的蜜源重新放回蜜源集合中,可以继续参与选择.

OV交叉操作:OV交叉操作采用POX交叉算子^[27],操作过程如下:1)对于待加工工件集 J ,将其中的待加工工件随机分配到两个非空互补子集 Q_1 和 Q_2 中,从 X 中选出包含在 Q_1 子集内的工件,将选出工件内的所有工序复制到子代1,保持工序原本的位置不变,同理复制 X_{best} 和 Q_2 子集内工件中的所有工序到子代2;2)复制在 X 中且包含在 Q_2 子集工件里的所有工序,保持原有顺序依次插入到子代1的空缺中,由此得到完整且合法的子代1,由 X_{best} 和 Q_1 做类似操作可得到子代2;3)从子代1和子代2中选择较优的一个作为交叉后的蜜源.图6给出了一个生成子代1的示例.

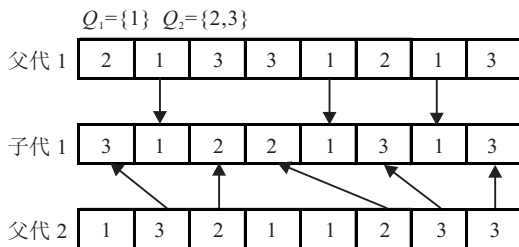


图6 OV交叉示意

UV交叉操作:UV交叉操作选择简单的两点交叉,操作过程如下:将两个父代复制到子代中,然后随机选择两个工件,互换两子代中两工件对应的FMU以及对应的加工设备,保持互换的加工设备顺序不变;利用贪婪选择从两个子代中得到最终交叉后的新蜜源.

MV交叉操作:MV的交叉操作采用RPX交叉算子^[28],具体实施步骤如下:先将父代1和父代2分别复制到子代1和子代2中;然后随机生成一个由0或1组成的字符串rand,字符串长度等于工序总数;遍历字符串rand,若rand中第 p 个元素等于0,则互换子代MV中位于位置 p 的加工设备索引号.此交叉操作可能会导致子代MV向量中出现非法值,因此在交叉完成之后需要检查加工设备是否在对应该FMU的可用设备集中,若不在,则从可用设备集中随机选择一个加工设备进行替换.最后,同样利用贪婪选择从子代1和子代2中选择一个较优的作为交叉后的新蜜源.图7展示了MV的交叉操作.

OV变异操作:在OV中随机选择一个元素 w ,将其替换为一个不等于 w 的工件号 p ;然后在OV中所有等于 p 的元素中随机选择一个替换成 w ,这样变异后得到的OV依然合法.

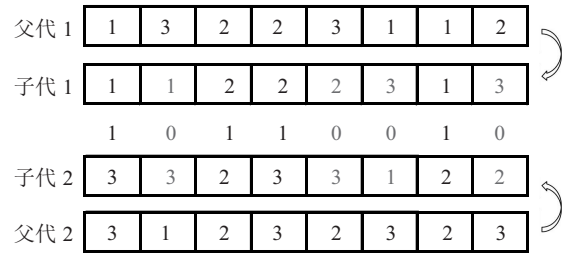


图7 MV交叉示意

UV、MV变异操作:UV及MV变异操作采用简单的替换变异,即随机选择一个要变异的位置,然后从可行范围内随机选择一个新值替换即可.需要注意的是,在UV进行变异后,由于每个FMU中设备数目及设备加工能力不同,需要检查和调整MV向量以确保变异后得到的是合法解.

由于设计的交叉、变异操作算子较多,若在每次的引领蜂搜索操作中都执行所有操作算子,则会增加算法耗时,而且有可能在后续的操作算子中破坏前面已得到的优良蜜源.为解决前述问题,本文设计了简单有效的操作算子选择方法,使用方法如下.

Step 1: 利用父代选择操作得到两父代蜜源;

Step 2: 进行OV交叉操作得到新蜜源,评价新蜜源,若新蜜源优于当前蜜源,则转Step 8;

Step 3: 进行UV交叉操作得到新蜜源,评价新蜜源,若新蜜源优于当前蜜源,则转Step 8;

Step 4: 进行MV交叉操作得到新蜜源,评价新蜜源,若新蜜源优于当前蜜源,则转Step 8;

Step 5: 进行OV变异操作得到新蜜源,评价新蜜源,若新蜜源优于当前蜜源,则转Step 8;

Step 6: 进行UV变异操作得到新蜜源,评价新蜜源,若新蜜源优于当前蜜源,则转Step 8;

Step 7: 进行MV变异操作得到新蜜源,评价新蜜源,若新蜜源优于当前蜜源,则转Step 8,否则转Step 9;

Step 8: 用新蜜源替换当前蜜源,置蜜源的尝试次数为0;

Step 9: 蜜源的尝试次数加1.

2.3.4 跟随蜂蜜源选择操作

跟随蜂对蜜源的选择通常采用轮盘赌方法,该方法是一种相对贪婪的选择方法,很难选中稍差的蜜源,实际中这些蜜源仍可能含有有用的信息,抛弃它们会影响算法的全局收敛能力.排序选择策略^[29]被证明具有良好的效果,并且操作简单、易于实现.本文使用排序选择策略实现跟随蜂蜜源选择操作,实施步骤为先根据式(12)计算种群中所有蜜源的适应度值,然后按照适应度值由大到小对所有蜜源进行排序,最后按照下述公式计算排序后第 k 个蜜源被选择的概

率,即

$$P_k = \frac{1}{SN} + a(t) \frac{SN + 1 - 2k}{SN(SN + 1)}, k = 1, 2, \dots, SN; \quad (15)$$

$$a(t) = 0.2 + \frac{3t}{4gen_{max}}, t = 1, 2, \dots, gen_{max}. \quad (16)$$

其中:SN为蜜源个数; $a(t)$ 为自适应参数, t 为当前迭代次数; gen_{max} 为最大迭代次数.利用该策略,由于在进化初期阶段, $a(t)$ 值较小,蜜源被选择的概率更均匀,可以保持种群的多样性;到进化后期阶段,较大的 $a(t)$ 值有助于克服由于种群多样性降低、竞争力降低而导致的进化停滞.

2.3.5 跟随蜂搜索操作

在跟随蜂的搜索操作中,使用基于关键路径的局部搜索以加强算法的局部搜索能力.从解码方法中可以看出,全局的完工时间等于所有FMU本地完工时间中的最大值,显然本地完工时间最大的FMU是其中的关键FMU,即关键路径.缩短关键FMU的本地完工时间,即可直接缩短全局完工时间.缩短关键FMU本地完工时间有两条途径:1)工件转移搜索,即将安排在关键FMU上加工的工件交给其他FMU来加工;2)自身变异搜索,即在关键FMU自身内部进行工序排序调整、加工设备重新分配操作.

利用上述两种搜索策略在搜索新蜜源的过程中可能遇到的情形如图8所示.在局部搜索算法中,基于以下原则处理遇到的所有情形.

- 情形1** 新蜜源缩短了全局完工时间,用新蜜源替换当前蜜源;
- 情形2** 新蜜源增长了全局完工时间,抛弃新蜜源;
- 情形3** 新蜜源保持全局完工时间未变,以一定概率选择用新蜜源替换当前蜜源.

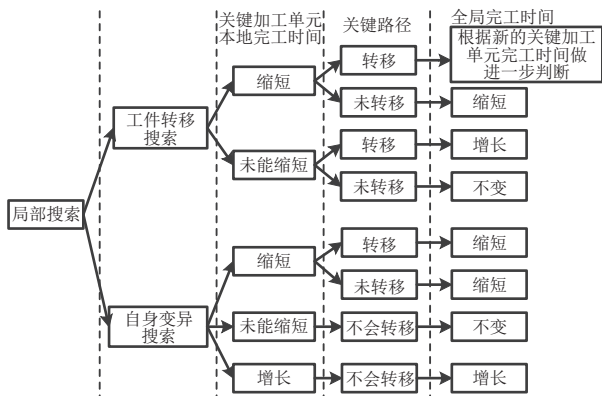


图8 局部搜索操作

基于以上原则设计局部搜索操作,其具体流程描述如下.

Step 1: 初始化.

Step 1.1: 获取当前蜜源 x ,全局完工时间 C_{max} ;

Step 1.2: 获取当前关键FMU U_c ,其他的非关键FMU集合 Ω ;

Step 1.3: 设置 $iter = 0, a = 0$.

Step 2: 选择搜索途径.

Step 2.1: 如果 $a > iter/5$,则转到Step 7,否则 $iter = iter + 1$;

Step 2.2: 产生一个随机数 $r \in (0, 1)$,如果 r 小于概率参数 R ,则进入Step 3,否则进入Step 4.

Step 3: 工件转移搜索.

随机选择一个由 U_c 加工的工件 J ,将工件 J 安排到 Ω 中本地完工时间最短的FMU中加工,得到新蜜源 X' .

Step 4: 自身变异搜索.

针对 X 中与 U_c 相关的工序与设备,通过调换工序加工顺序、重新选择加工设备得到新蜜源 X' ,实施步骤可参考上述OV、MV变异操作.

Step 5: 新蜜源贪婪选择.

Step 5.1: 计算新蜜源 X' 的全局完工时间 C'_{max} ;

Step 5.2: 若 $C'_{max} < C_{max}$,则转到Step 5.4,若 $C'_{max} = C_{max}$,则转到Step 5.3,其他转Step 5.5;

Step 5.3: 产生一个随机数 $r \in (0, 1)$,如果 $r < 0.5$,则转到Step 5.4,否则转到Step 5.5;

Step 5.4: 用新蜜源 X' 替换掉当前蜜源 X ,更新 U_c 及 Ω ,设置 $a = 0$;

Step 5.5: 设置 $a = a + 1$.

Step 6: 判断迭代条件.

若 $iter$ 达到最大迭代次数 $iter_{max}$,则进入Step 7,否则进入Step 2.

Step 7: 终止.

终止迭代.

2.3.6 侦查蜂搜索操作

判断是否存在尝试次数超出限定值的蜜源,若存在,则选择一个蜜源(如果有多个,则尽量选择尝试次数最多的,如果有多个选择,则随机选择一个)进行替换操作,用来替换的新蜜源采用随机生成的方法得到.

3 实例验证及结果分析

为验证IABC求解DFJSP的性能,进行如下操作:首先通过扩展FJSP的标准测试算例得到适用于DFJSP的测试算例;然后,采用正交试验法对IABC中的参数进行优化配置;最后,将最新研究成果^[30]中提出的HGA以及扩展后适用于求解DFJSP的TA(Threshold accepting)^[31-33]作为对比算法来验证IABC的求解性能.此外,将IABC中的局部搜索算子

替换为引领蜂的搜索操作并命名为DABC,以DABC作为对比算法来验证IABC中提出的局部搜索的有效性.

3.1 算例生成

当前并没有标准测试算例可用来测试DFJSP,因此本文以Brandimarte^[34]柔性作业车间调度问题标准算例中的MK01, Mk04, MK09, Mk12, Mk15为基础,针对每个标准算例分别考虑3,4,5个FMU的情况,并生成15个新测试算例以供使用.新生成的算例命名为“原名称_FMU数目”,例如,以MK01算例为基础,考虑有3个FMU生成的新算例,命名为MK01_3.生成新算例时所使用的参数如表2所示.

表2 生成算例所用参数

算例名称	工件总数	单工件工序数	FMU总数	单FMU中设备总数	单工序加工时间
MK01	10	[5,7]	{3,4,5}	[5,7]	[1,7]
MK04	15	[3,10]	{3,4,5}	[6,10]	[1,10]
MK09	20	[10,15]	{3,4,5}	[8,12]	[5,20]
MK12	30	[5,10]	{3,4,5}	[8,12]	[10,30]
MK15	30	[8,12]	{3,4,5}	[12,17]	[10,30]

具体生成步骤以生成新算例MK01_3为例:工件数目为10,单个工件中的工序数目分别在[5,7]内随机选取;FMU数目为3,每个FMU内的加工设备数分别在[5,7]内随机选取;生成工序的加工时间,遍历所有加工设备,以50%的概率选择当前加工设备作为可用加工设备,加工时间为整数且在[1,7]内随机选取.需要注意的是每个FMU都需要具备加工所有工序的能力,因此,当出现FMU内所有设备都不能加工某个工序的情况时,随机选择一台设备成为可加工设备.

3.2 参数设计

在IABC中,影响其性能的参数主要有4个:蜂群规模SN,最大尝试次数limit,局部搜索最大迭代次数 $iter_{max}$,局部搜索概率参数R.采用正交试验法对参数进行优化配置,每个参数的水平定为5个水平,如表3所示.

表3 因素水平配置表

参数水平	参数			
	SN	limit	$iter_{max}$	R
1	50	3	10	0.3
2	70	5	20	0.4
3	100	8	30	0.5
4	130	10	40	0.6
5	150	15	50	0.7

选用正交表 $L_{25}(5^4)$,以MK04_3作为测试算例,算法运行终止条件设置为迭代次数达到200次.为

避免测试结果的随机性,每组实验运行30次,将30次运行结果的平均值作为最终实验结果列在表4中.表4中实验结果最优值是40.33,对应的参数设置为 $SN = 130, limit = 15, iter_{max} = 40, R = 0.3$,此时IABC性能最佳.

表4 参数设计实验结果及分析

实验号	参数				实验结果
	SN	limit	$iter_{max}$	R	
1	70	5	40	0.4	44.37
2	150	3	20	0.4	44.50
3	130	3	30	0.5	44.30
4	50	5	50	0.5	44.23
5	100	3	40	0.6	44.97
6	70	8	30	0.6	44.23
7	130	5	20	0.7	45.20
8	150	5	10	0.6	45.70
9	150	10	40	0.5	42.47
10	130	10	50	0.6	42.00
11	100	10	10	0.7	44.90
12	70	10	20	0.3	43.53
13	70	3	50	0.7	45.20
14	130	8	10	0.4	43.73
15	50	8	40	0.7	44.50
16	50	3	10	0.3	46.90
17	100	5	30	0.3	43.47
18	150	15	30	0.7	42.40
19	50	10	30	0.4	43.07
20	100	15	50	0.4	41.13
21	70	15	10	0.5	44.03
22	130	15	40	0.3	40.33
23	150	8	50	0.3	41.23
24	50	15	20	0.6	44.07
25	100	8	20	0.5	43.13

3.3 实验结果及结论

算法运行环境为: Intel Core I7-5700HQ CPU 2.70GHz, RAM 8GB, Windows 7, 64位操作系统, 编程语言为C#. DABC参数设置与IABC一致, HGA参数设置与文献[30]中一致, TA中参数由正交试验获得. 为避免随机性,每个算例将算法重复运行30次,得到30个结果值. IABC终止条件设置为最大迭代次数200次,为公平比较算法的性能, DABC、HGA、TA的终止条件设置为IABC迭代200次的平均运行时间. 最终实验结果如表5所示. 表5中Best为30次运行中的最优值, Avg为平均值, CPU为程序运行时间.

从表5中各算例结果的最优值和平均值可以看出, IABC在大部分算例上都表现得最为优异, HGA次之, 由此表明IABC在求解DFJSP时具有优良的性能, 同时IABC与DABC的对比也验证了局部搜索算子的有效性. 为了验证算法性能是否确实存在统计意义上的差异, 利用实验结果中的数据进行Wilcoxon测试(置信水平为0.05), 实验结果如表6所示.

表 5 实验结果

算例	CPU/s	IABC		DABC		HGA		TA	
		Best	Avg	Best	Avg	Best	Avg	Best	Avg
MK01_3	10	18	21.3	19	23.83	18	21.37	19	21.67
MK04_3	18	34	40.33	42	45.7	35	41.83	43	48.50
MK09_3	74	173	187.03	202	213.43	176	208.27	206	220.30
MK12_3	68	222	235.97	247	261.73	225	231.5	251	260.50
MK15_3	101	236	251.97	259	270.8	243	254.23	264	272.83
MK01_4	13	17	19.67	20	23.17	17	20.37	22	24.57
MK04_4	24	32	37.73	35	39.47	32	37.3	37	41.73
MK09_4	98	161	172.4	184	194.87	166	175.67	187	196.93
MK12_4	89	193	205.73	218	225.83	201	209.63	224	235.73
MK15_4	135	216	230.43	233	251.83	227	239.5	241	252.20
MK01_5	17	17	19.27	20	22.9	18	20.33	20	24.27
MK04_5	30	29	34.17	32	40.6	32	35.97	36	41.77
MK09_5	124	150	162	166	175.17	151	160.7	172	182.03
MK12_5	112	182	196.37	195	200.87	186	197	201	208.80
MK15_5	171	209	223.9	230	238.87	214	224.13	229	247.83

表 6 Wilcoxon 测试结果

算法	Best			Avg		
	R ⁺	R ⁻	p-value	R ⁺	R ⁻	p-value
IABC vs DABC	0	15	0.001	0	15	0.001
IABC vs HGA	0	12	0.002	3	12	0.036
IABC vs TA	0	15	0.001	0	15	0.001

表 6 中 p 值均小于 0.05, 说明 IABC 在最优值和均值两指标上均显著好于其他 3 种算法。

4 结 论

本文研究了利用改进型人工蜂群算法求解分布式柔性作业车间调度问题。首先, 建立了分布式柔性作业车间调度模型, 然后, 针对问题特点改进了人工蜂群算法, 包括设计编码方案、种群初始化方法以及交叉和变异搜索操作, 此外基于关键路径思想设计了局部搜索算子以加强算法的局部搜索能力; 最后, 设计对比实验验证了算法性能, 结果表明改进人工蜂群算法能有效求解该问题, 而且局部搜索算子明显提升了算法的寻优能力。

就目前而言, 关于分布式柔性作业车间调度问题这一研究课题, 还有许多欠缺有待进一步深入研究和探索, 大致可以在以下几个方向展开研究: 1) 寻找新的算法或者混合算法求解该问题; 2) 研究动态环境下的分布式柔性作业车间问题; 3) 考虑其他更能反映分布式调度特征的优化目标, 或者多目标。

参考文献(References)

[1] 吴秀丽, 张志强, 杜彦华, 等. 改进细菌觅食算法求解柔性作业车间调度问题[J]. 计算机集成制造系统, 2015, 21(5): 1262-1270.
(Wu X L, Zhang Z Q, Du Y H, et al. Improved bacteria foraging optimization algorithm for flexible job shop[J]. Computer Integrated Manufacturing Systems,

2015, 21(5): 1262-1270.)
 [2] Xu Y, Wang L, Wang S Y, et al. An effective teaching-learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time[J]. Neurocomputing, 2015, 148: 260-268.
 [3] Wu X, Sun Y. A green scheduling algorithm for flexible job shop with energy-saving measures[J]. J of Cleaner Production, 2018, 172: 3249-3264.
 [4] Lei D, Li M, Wang L. A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold[J]. IEEE Trans on Cybernetics, 2018, 99: 1-13.
 [5] Lei D, Zheng Y, Guo X. A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption[J]. Int J of Production Research, 2017, 55(11): 3126-3140.
 [6] 王凌, 邓瑾, 王圣尧. 分布式车间调度优化算法研究综述[J]. 控制与决策, 2016, 31(1): 1-11.
(Wang L, Deng J, Wang S J. Survey on optimization algorithms for distributed shop scheduling[J]. Control and Decision, 2016, 31(1): 1-11.)
 [7] Chan F T S, Chung S H, Chan P L Y. An adaptive genetic algorithm with dominated genes for distributed scheduling problems[J]. Expert Systems with Applications, 2005, 29(2): 364-371.
 [8] Chan F T S, Chung S H, Chan P L Y. Application of genetic algorithms with dominant genes in a distributed scheduling problem in flexible manufacturing systems[J]. Int J of Production Research, 2006, 44(3): 523-543.
 [9] Giovanni L D, Pezzella F. An improved genetic algorithm for the distributed and flexible job-shop scheduling problem[J]. European J of Operational Research, 2010, 200(2): 395-408.
 [10] 黄英杰, 姚锡凡. 基于目标级联法和粒子群算法的柔性分布式车间调度[J]. 中南大学学报: 自然科学版, 2012, 43(1): 151-158.
(Huang Y J, Yao X F. Planning and scheduling of multiple flexible-shops based on analytical target cascading and particle swarm optimization[J]. J of Central South University: Science and Technology, 2012, 43(1): 151-158.)
 [11] 杨江波, 陈友玲, 曹楠. 面向柔性作业分布式车间的分层调度模型研究[J]. 计算机工程与应用, 2014, 50(23): 239-244.
(Yang J B, Chen Y L, Cao N. Research on grading scheduling model of flexible job distributed workshop[J]. Computer Engineering and Applications, 2014, 50(23): 239-244.)
 [12] Ziaee M. A heuristic algorithm for the distributed and flexible job-shop scheduling problem[J]. J of Supercomputing, 2014, 67(1): 69-83.
 [13] Lu P H, Wu M C, Tan H, et al. A genetic algorithm embedded with a concise chromosome representation for distributed and flexible job-shop scheduling problems[J]. J of Intelligent Manufacturing, 2018, 29(1): 19-34.

- [14] Karaboga D. An idea based on honey bee swarm for numerical optimization[R]. Kayseri: Computer Engineering Department, Erciyes University, 2005.
- [15] Gao W F, Liu S Y. A modified artificial bee colony algorithm[J]. Computers & Operations Research, 2012, 39(3): 687-697.
- [16] 李田来, 刘方爱, 王新华. 基于分治策略的改进人工蜂群算法[J]. 控制与决策, 2015, 30(2): 316-320.
(Li T L, Liu F A, Wang X H. Modified artificial bee colony algorithm based on divide-and-conquer strategy[J]. Control and Decision, 2015, 30(2): 316-320.)
- [17] 周新宇, 吴志健, 邓长寿, 等. 一种邻域搜索的人工蜂群算法[J]. 中南大学学报: 自然科学版, 2015, 46(2): 534-546.
(Zhou X Y, Wu Z J, Deng C S, et al. Neighborhood search based artificial bee colony algorithm[J]. J of Central South University: Science and Technology, 2015, 46(2): 534-546.)
- [18] Lei D M, Guo X P. Scheduling job shop with lot streaming and transportation through a modified artificial bee colony[J]. Int J of Production Research, 2013, 51(16): 4930-4941.
- [19] Gupta M, Sharma G. An efficient modified artificial bee colony algorithm for job scheduling problem[J]. Int J of Soft Computing & Engineering, 2012, 1(6): 291-296.
- [20] Pan Q K, Wang L, Li J Q, et al. A novel discrete artificial bee colony algorithm for the hybrid flowshop scheduling problem with makespan minimisation[J]. Omega, 2014, 45(2): 42-56.
- [21] Tasgetiren M F, Pan Q K, Suganthan P N, et al. A discrete artificial bee colony algorithm for the no-idle permutation flowshop scheduling problem with the total tardiness criterion[J]. Applied Mathematical Modelling, 2013, 37(10/11): 6758-6779.
- [22] Wang L, Zhou G, Xu Y, et al. An effective artificial bee colony algorithm for the flexible job-shop scheduling problem[J]. Int J of Advanced Manufacturing Technology, 2012, 60(1/2/3/4): 303-315.
- [23] Gao K Z, Suganthan P N, Pan Q K, et al. An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time[J]. Expert Systems with Applications, 2016, 65: 52-67.
- [24] Li X, Peng Z, Du B, et al. Hybrid artificial bee colony algorithm with a rescheduling strategy for solving flexible job shop scheduling problems[J]. Computers & Industrial Engineering, 2017, 113: 10-26.
- [25] Ling Wang, Gang Zhou, Ye Xu, et al. A hybrid artificial bee colony algorithm for the fuzzy flexible job-shop scheduling problem[J]. Int J of Production Research, 2013, 51(12): 3593-3608.
- [26] 刘三阳, 张平, 朱明敏. 基于局部搜索的人工蜂群算法[J]. 控制与决策, 2014, 29(1): 123-128.
(Liu S Y, Zhang P, Zhu M M. Artificial bee colony algorithm based on local search[J]. Control and Decision, 2014, 29(1): 123-128.)
- [27] 张国辉, 高亮, 李培根, 等. 改进遗传算法求解柔性作业车间调度问题[J]. 机械工程学报, 2009, 45(7): 145-151.
(Zhang G H, Gao L, Li P G, et al. Improved genetic algorithm for the flexible job-shop scheduling problem[J]. J of Mechanical Engineering, 2009, 45(7): 145-151.)
- [28] 张静, 王万良, 徐新黎, 等. 基于改进粒子群算法求解柔性作业车间批量调度问题[J]. 控制与决策, 2012, 27(4): 513-518.
(Zhang J, Wang W L, Xu X L, et al. Improved particle swarm algorithm for batch splitting flexible job shop scheduling[J]. Control and Decision, 2012, 27(4): 513-518.)
- [29] Bao L, Zeng J. Comparison and analysis of the selection mechanism in the artificial bee colony algorithm[C]. Ninth Int Conf on Hybrid Intelligent Systems. Shenyang: IEEE Computer Society, 2009: 411-416.
- [30] Chang H C, Liu T K. Optimisation of distributed manufacturing flexible job shop scheduling by using hybrid genetic algorithms[J]. J of Intelligent Manufacturing, 2017, 28(8): 1973-1986.
- [31] 庞哈利. 柔性Job shop集成化计划调度模型及其求解算法[J]. 控制与决策, 2003, 18(1): 34-39.
(Pang H L. Integrated planning and scheduling model and hybrid heuristic algorithm for flexible job shop[J]. Control and Decision, 2003, 18(1): 34-39.)
- [32] 庞哈利, 郑秉霖, 徐心和. 多目标Flow Shop调度问题的改进TA求解算法[J]. 系统工程理论与实践, 2000, 20(5): 62-65.
(Pang H L, Zheng B L, Xu X H. An improved TA algorithm for multiple criteria flow shop scheduling problem[J]. System Engineering—Theory & Practice, 2000, 20(5): 62-65.)
- [33] Duecka G, Scheuera T. Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing[J]. J of Computational Physics, 1990, 90(1): 161-175.
- [34] Brandimarte P. Routing and scheduling in a flexible job shop by tabu search[J]. Annals of Operations Research, 1993, 41(3): 157-183.

作者简介

吴锐(1989—), 男, 博士生, 从事车间调度及智能优化算法的研究, E-mail: wurui@whut.edu.cn;

郭顺生(1963—), 男, 教授, 博士生导师, 从事智能制造、智能决策技术等研究, E-mail: guoshunsheng@whut.edu.cn;

李益兵(1978—), 男, 教授, 博士, 从事智能制造与企业信息化等研究, E-mail: ahlyb@whut.edu.cn;

王磊(1988—), 男, 博士, 从事制造业信息化、制造资源配置的研究, E-mail: wanglei9455@126.com;

许文祥(1991—), 男, 博士生, 从事制造业信息化、制造过程监控的研究, E-mail: xiang1054097606@163.com.

(责任编辑: 闫妍)