控制与决策
Control and Decision

基于改进蛙跳算法的分布式两阶段混合流水车间调度

雷德明， 王甜

引用本文:
雷德明, 王甜. 基于改进蛙跳算法的分布式两阶段混合流水车间调度[J]. 控制与决策, 2021, 36(1): 241−248.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2019.0472

基于改进蛙跳算法的分布式两阶段混合流水车间调度

雷德明†, 王 甜

(武汉理工大学 自动化学院，武汉 430070)

摘 要: 针对考虑顺序相关准备时间的分布式两阶段混合流水车间调度问题,提出一种改进的蛙跳算法以同时最小化拖后工件数和最大完成时间. 该算法通过启发式方法和随机方法对种群进行初始化,采取基于种群和记忆的种群划分方法,同时给出模因组质量评价方法,并根据模因组质量将所有模因组划分为最优模因组、最差模因组和其他模因组,每种类型的模因组分别采取不同的搜索策略,并分配不同的搜索次数,其中最优模因组不参与种群划分. 选用一种多目标经典算法和两种近5年提出的算法作为对比算法,并与改进蛙跳算法的变体进行比较以验证模因组搜索新策略的有效性. 通过对大量实例的计算实验结果表明,模因组搜索新策略有效,改进蛙跳算法能有效求解分布式两阶段混合流水车间调度问题.

关键词: 分布式调度；两阶段；混合流水车间；准备时间；蛙跳算法；模因组分类

中图分类号: TP18 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2019.0472 **开放科学(资源服务)标识码(OSID):**

引用格式: 雷德明,王甜. 基于改进蛙跳算法的分布式两阶段混合流水车间调度[J]. 控制与决策, 2021, 36(1): 241-248.

An improved shuffled frog leaping algorithm for the distributed two-stage hybrid flow shop scheduling

LEI De-ming†, WANG Tian

(College of Automation，Wuhan University of Technology，Wuhan 430070，China)

Abstract: To solve the distributed two-stage hybrid flow shop scheduling problem (DTHFSP) with sequence-dependent setup times, an improved shuffled frog leaping algorithm is proposed to minimize the number of tardy jobs and makespan simultaneously. The heuristic method and random method are adopted to initialize the population, population division is done by using population and memory, and then all memeplexes are divided into a best memeplex, a worst memeplex and other memeplexes according to the quality of memeplexes. Different search methods and the number of searches are applied to different kinds of memeplexes and the best memeplex is excluded from the population division. A multi-objective classical algorithm and two algorithms proposed in recent five years are selected as the comparative algorithms, and a variation of the improved shuffled frog leaping algorithm is used for comparison in order to verify the effectiveness of the new search strategy of memeplexes. The comparative analysis based on extensive instances shows that the new search strategy of memeplexes is effective and the proposed algorithm can solve the DTHFSP effectively.

Keywords: distributed scheduling; two-stage; hybrid flow shop; setup times; shuffled frog leaping algorithm; memeplexes classification

0 引 言

全球化背景下,企业需要在不同地方建立工厂以快速响应市场变化和客户需求,分布式制造已成为一种常见的制造模式. 作为分布式制造的重要环节,分布式车间调度突破了传统调度的单工厂环境,是指在加工能力和环境等相同或各异的多个工厂间资源分配和制造任务的排序调度. 与单工厂调度相比,分布式调度子问题更多且子问题间耦合关系更强,已成为目前调度研究的热点问题.

混合流水车间调度问题(hybrid flowshop scheduling problem, HFSP)是一类广泛存在于造纸、纺织、制药、化工和半导体等制造业的柔性调度问题. 目前,HFSP在单工厂环境下的研究取得了很大进展,出现了大量的研究成果[1-11]. 在多工厂环境下,分布式混合流水车间调度问题(distributed hybrid flowshop scheduling problem, DHFSP)未受到研究者

收稿日期: 2019-04-15；**修回日期:** 2019-07-30.
基金项目: 国家自然科学基金项目(61573264).
†通讯作者. E-mail: deminglei11@163.com.

的重视,研究结果较少. Ying等[12]提出了一种混合整数线性规划和自整定迭代贪婪(IG)算法以解决具有多处理器任务的DHFSP. 蔡劲草等[13]以最小化最大完成时间和最大延迟时间为目标,提出一种双变邻域搜索算法解决两阶段DHFSP. DHFSP的特殊情形——分布式流水车间调度问题(distributed flowshop scheduling problem, DFSP)的研究取得了一些进展[14-20].

DHFSP和DFSP的现有研究以最小化makespan为主,较少同时优化多个彼此冲突的目标,如拖后工件数和makespan,一些常见的加工约束如顺序相关调整时间(SDST)也未结合到DHFSP中. 本文研究的DTHFSP是以最小化拖后工件数和makespan为目标,考虑SDST,且各工厂第1阶段均为单机,第2阶段为并行机的一类两阶段DHFSP. 目前大部分研究主要采用智能优化方法对问题求解,以IG的应用为主,其他智能算法如蛙跳算法(shuffled frog-leaping algorithm, SFLA)很少用来解决DHFSP和DFSP.

SFLA结合了文化基因算法和粒子群算法的优点,具有概念简单、参数少、计算速度快、全局搜索寻优能力强、易于实现等特点. 目前,SFLA在制造系统调度方面有广泛应用[21-29],只是SFLA在分布式调度和DHFSP方面的应用研究较少,因此有必要探讨SFLA在DHFSP方面的搜索优势.

本文针对DTHFSP,应用一种改进的SFLA以最小化拖后工件数和makespan. 通过启发式方法和随机方法对种群进行初始化,给出模因组质量评价方法,并根据模因组质量将所有模因组划分为最优模因组、最差模因组和其他模因组,每种类型的模因组分别采取不同的搜索策略. 最后通过大量计算实验验证SFLA在DTHFSP方面的搜索优势.

1 问题描述

DTHFSP由n个独立的工件和f个工厂组成,工件集合$J=\{J_1,J_2,\dots,J_n\}$,工厂集合$F=\{F_1,F_2,\dots,F_f\}$. 工厂F_l包含一个两阶段混合流水车间,其中第1阶段为单机,第2阶段由m_l台同速并行机组成. 工件J_i在工厂F_l第k阶段的加工时间为p_{ilk},其交货期为d_i. 工件的准备时间依赖于工件的加工顺序,如果工件J_i为工厂F_l第k阶段中机器上首次加工的工件,则准备时间为u_{0ilk},如果工件J_j、J_i在工厂F_l的第k阶段的同一机器上依次加工,则工件J_i的准备时间为u_{jilk}.

DTHFSP还需满足如下约束条件:一个工件同一时刻最多只能在一台机器上加工,一台机器同一时刻最多只能加工一个工件,所有工件的加工过程不能中断,所有机器在任意时刻都可用等.

DTHFSP包含3个子问题:工厂分配、第2阶段机器分配和调度,其中工厂分配将工件分配到合适的工厂,机器分配为工厂内的工件分配同一工厂的机器. 3个子问题间具有强耦合关系,例如,工厂分配直接决定机器分配和调度的内容,工厂分配的好坏取决于机器分配和调度的质量.

由于每个工厂第2阶段由同速并行机组成,工件在这些并行机上的分配通常根据机器最早可开始加工时间来为工件选择合适的机器,这样只需对工厂分配和调度子问题进行优化,从而降低了子问题间的耦合关系.

根据上述分析,问题的目标为给每个工件分配合适的机器并确定各阶段机器上工件的加工顺序以最小化如下目标:

$$C_{\max}=\max_{i=1,2,\dots,n}\{C_i\}, \tag{1}$$

$$n_t=|\{J_i|C_i>d_i\}|. \tag{2}$$

其中:C_i为工件i的完成时间,$C_{\max}$为最大完成时间,n_t为拖后工件数.

2 基于改进SFLA的DTHFSP

蛙跳算法是一种仿生物学的智能优化算法,其主要步骤包括种群划分、模因组搜索和种群重构. 通常,各模因组的搜索彼此独立,并且所有模因组的搜索方式和过程基本一样,很少根据模因组质量对模因组进行再分组,然后根据各模因组特点采取各种独特的搜索方式. 基于上述思想,提出一种改进SFLA,解决以makespan和拖后工件数为目标的DTHFSP.

2.1 编码和解码

由于第2阶段机器分配采用启发式规则,只需对DTHFSP的工厂分配和调度两个子问题进行编码,为此采用双串编码方式. 其中:第1个串为工厂分配串$[h_1,h_2,\dots,h_n]$,第2个串为调度串$[q_1,q_2,\dots,q_n]$. 工厂分配串中,$h_i\in\{1,2,\dots,f\}$表示工件J_i分配到工厂h_i,调度串为实数串,q_i为工件J_i的调度值,调度值越小,该工件的加工优先级越高.

解码过程如下:首先,根据工厂分配串确定每个工件所在的工厂,从而确定每个工厂第1阶段机器加工的所有工件;然后,针对每个工厂第1阶段的唯一机器,对该机器加工的所有工件根据其对应的调度值q_v进行升序排序,得到工件排列;从工件排列的第1个工件开始进行第1阶段的加工,排列的所有工件中先完成第1阶段加工的工件进入第2阶段,从并行

机中选择一台最早可开始加工的机器,若多台机器的最早可开始时间一样,则选择机器号最小的机器,完成工件第2阶段的加工.

DTHFSP的特点在于它只有两个阶段,且第1阶段为单机,机器分配只存在于第2阶段,故采用工厂分配串与调度串并结合启发式方法进行第2阶段机器分配,以降低求解难度,保证求解质量.

2.2 初始种群和种群划分

每个工厂的第1阶段为单机,第2阶段由并行机组成,这样导致工件在第1阶段的加工顺序直接影响其在第2阶段的加工顺序,从而影响最终完成时间. 根据问题的这一特点,提出一种启发式方法产生一半的初始种群,具体描述如下:随机产生调度串,并根据调度串确定所有工件的排列,将排列中的前 f 个工件依次分配到工厂 $1,2,\ldots,f$,从排列的第 $f+1$ 个工件开始,按照解码的机器分配规则为每个工件选择第1阶段最早可开始加工的机器,将工件分配到该机器所在的工厂,由此产生工厂分配串. 剩下的 $N/2$ 个个体随机产生.

由于各工厂第1阶段的机器数总是小于等于第2阶段的机器数,将工件均匀分配到各工厂的第1阶段机器上将产生一些较优的解,从而得到一个较优的初始种群.

对于解决多目标问题,现有SFLA很少对个体质量和整个模因组质量进行评价. 本文给出个体的质量定义,并根据个体质量进行种群划分,在种群划分后根据模因组质量对所有模因组分类.

每个个体的质量定义为种群中受该个体支配的其他解的数量. 若解 x' 和 x 满足如下条件:$C_{\max}(x') < C_{\max}(x)$ 且 $n_t(x') \leqslant n_t(x)$,或 $C_{\max}(x') \leqslant C_{\max}(x)$ 且 $n_t(x') < n_t(x)$,则 x' 支配 x 或 x 受 x' 支配; 若上述条件不成立,则称 x 不受 x' 支配.

引入记忆解集 Ω,其最大容量为 V,用来保留搜索过程中未进入种群或种群中被替换的解,这些解质量较优,将它们直接丢弃会导致计算资源浪费,因此将它们放入 Ω,使它们有机会通过种群划分再次进入种群. 初始 Ω 由初始种群中质量最好的 V 个解构成.

本文提出一种新的种群划分策略,利用集合 Ω 和种群 P 构建 s 个模因组,具体过程如下:每个模因组的第1个解来自 Ω,即从 Ω 中随机选择两个解,在两者中选择一个非劣排序所得rank值较小的解,若两个解rank一样,则选择拥挤距离较大的解进入模因组;模因组其他成员来自种群,依次从种群 P 中随机选择两个解,采用上述基于rank值和拥挤距离的方法选择较好解进入模因组,重复进行直到所有模因组内解的总数等于 N 为止.

构建 s 个模因组后,计算每个模因组的总质量,它等于组内所有个体的质量之和. 根据模因组质量,将 s 个模因组分成3类:第1类由质量最大的一个模因组组成,第2类由质量最差的模因组组成,其他模因组为第3类. 分类的目的在于根据每类模因组特点采用相应的搜索策略.

2.3 模因组搜索

模因组搜索是SFLA产生新解的主要方式,其主要内容包含优化对象的选择以及针对优化对象的全局或局部搜索. 通常,模因组内的最差解 x_w 为优化对象,也有研究选择最好解 x_b 作为优化对象[30-31],而优化对象的搜索过程在所有模因组都是一样或相似的,缺乏针对模因组特点而设计模因组搜索过程的相关研究. 本文在上述模因组分类基础上,对不同类的模因组采用不同的搜索策略,并给好的模因组分配更多的搜索次数,减少差的模因组的搜索次数,以保证算法向最优解可能的方向进行搜索,避免无效搜索带来的资源浪费.

模因组搜索过程描述如下.

step 1: 对于第1类模因组,重复执行如下步骤 μ_1 次:确定模因组内质量最大的个体 x_b,随机选择模因组内除 x_b 外的解 x 作为优化对象,执行全局搜索GS,对 x_b 和对整个种群质量最大的解 x_g 分别进行局部搜索 β 次.

step 2: 对于第2类模因组,重复执行如下步骤 μ_2 次:确定模因组内质量最小的个体 x_w,针对第1类模因组质量最大的个体进行局部搜索 β 次.

step 3: 对于第3类模因组中的每个模因组,重复执行如下步骤 μ_3 次:确定模因组内质量最大的个体 x_b 和最差的个体 x_w,执行全局搜索GS.

step 1 ~ step 3 中:μ_1、μ_2、μ_3 为各类模因组搜索过程的循环次数,β 为局部搜索次数. 全局搜索GS描述如下:产生区间[0, 1]内服从均匀分布的随机数 α,如果 $\alpha < \theta$(其中 θ 为实数),则对调度串进行全局搜索,否则对工厂分配串执行全局搜索. 只有第1类模因组GS的对象为模因组内除 x_b 外随机选择的个体,第3类模因组GS的对象为 x_w.

两个串的全局搜索过程类似,下面以 x_w 为对象描述调度串的全局搜索过程:对于模因组的最好解 x_b 和最差解 x_w,从调度串中随机选择位置 k_1、k_2, $k_1 < k_2$, x_w 调度串中位于两个位置间的基因由 x_b 调度串中同样位置间的基因替代,得到新解 y,如

果y支配x_w,则用x_w更新Ω并且y替代x_w,否则用y更新记忆解集,利用种群内质量最大的个体x_g代替x_b,重复上述过程,若新解仍不能支配x_w,则随机初始化x_w的调度串. 以第1类模因组内x为对象的全局搜索过程与上述过程一致,只是需要将x_w更换为x.

记忆Ω在搜索过程中,对于解y,若Ω未满,则$\Omega = \Omega\bigcup\{y\}$,否则将$\Omega$中受$y$支配的所有个体剔除;然后判断$\Omega$是否已满,若未满则将$y$添加到$\Omega$中.

蛙跳算法通常只对组内最差个体进行更新,对组内最好个体不作处理,导致算法难以实现探索和利用的平衡. 为此,对模因组的x_b和种群的x_g执行基于3种邻域结构的局部搜索. 邻域结构swap过程如下:随机选择串中两个不同位置的基因并互换. 邻域结构insert过程如下:随机选择两个不同位置的基因,将后一个基因插入到前一个基因的位置前. 邻域结构invert过程如下:随机选择串中两个位置,将两位置间的染色体片段前后颠倒. 这3个邻域结构适用于调度串和工厂分配串.

局部搜索过程如下:对于解z,采用与全局搜索同样的概率选择方式选择调度串或工厂分配串之一进行局部搜索. 以调度串为例描述局部搜索过程如下:对于z的调度串,应用swap产生新解y,若y支配z,则利用z更新Ω并运用y代替z,否则根据y更新Ω;对z再执行insert,产生新解y',若y'支配z,则用z更新Ω且令y'成为新的z,否则根据y'更新Ω;对z最后应用invert产生新解y'',若y''支配z,则根据z更新Ω并用y''替换z,否则用y''更新Ω.

对于第2类模因组,因其质量较差,故并不直接针对该模因组内的任何解执行局部搜索,而是选择第1类模因组内质量最大的解进行局部搜索,若局部搜索产生的新解y支配该模因组的x_w,则用x_w更新Ω,并利用y代替x_w,否则根据y更新Ω,同时减少该类模因组的搜索次数.

如上所述,每一类模因组的优化对象和搜索过程各异,且模因组搜索的次数$\mu_1 \neq \mu_2 \neq \mu_3$,这样可以采用多种方式产生新解,充分利用质量较高的解,避免计算资源浪费在质量较差的解上,从而改善种群的多样性,提高搜索效率.

2.4 算法描述

改进SFLA的详细过程描述如下.

step 1: 设置参数(包括N和模因组数量s等),采用启发式方法和随机方法产生初始种群P.

step 2: 对种群P的所有解进行非劣排序和拥挤距离计算,确定初始记忆解集Ω.

step 3: 当早期搜索条件成立时,执行如下过程:

step 3.1: 种群划分以构建s个模因组;

step 3.2: 对每个模因组采用第3类模因组的方式执行μ次搜索;

step 3.3: 种群重构.

step 4: 种群划分.

step 5: 当早期搜索条件不成立时,执行如下过程直到终止条件成立:

step 5.1: 将所有模因组分成3类;

step 5.2: 执行每一类模因组的搜索;

step 5.3: 确定新的最优模因组并将其直接作为第1类;

step 5.4: 种群重构和种群划分.

step 6: 输出$P\bigcup\Omega$中的非劣解.

第2阶段种群划分的过程与第1阶段的区别在于:首先将最优模因组直接从种群中拷贝出来,成为划分后的最优模因组,划分前后最优模因组保持不变,即最优模因组不参与种群划分.

如上所述,整个算法的搜索包括两个阶段,第1阶段所有模因组采用同样方式进行全局搜索,第2阶段在不同类模因组内运用不同的方式产生新解. 此外,上述算法将第1类模因组排斥在种群重构之外,以维持第1类模因组良好的解的构成,并加强对第1类模因组的质量最大个体的局部搜索,实现探索与利用间的良好平衡.

3 仿真实验与分析

选用大量实例进行对比分析实验,通过Microsoft Visual Studio C++ 2017实现,运行于4.0 G RAM 2.20 GHz CPU的个人电脑.

3.1 测试实例,对比算法以及评价指标

应用38个测试实例进行计算实验,表1描述了实例的信息,其中$p_{ilk} \in [50, 70]$,$u_{jilk} \in [5, 10]$,交货期

$$d_i = \delta\{\max_{1\leqslant l\leqslant f}\{\max_{1\leqslant k\leqslant 2}\{p_{ilk}\}\}+ \max_{1\leqslant j\leqslant n}\{\max_{1\leqslant l\leqslant f}\{\max_{1\leqslant k\leqslant 2}\{u_{jilk}\}\}\}\},$$

$$\delta \in [1, n/f + 1].$$

表1 实例的相关信息

实例	n	实例	n	实例	F	m_f
1, 11	30	7, 17, 25, 31	90	1∼10	2	2, 4
2, 12	40	8, 18, 26, 32	100	11∼20	3	2, 3, 4
3, 13, 21	50	9, 19, 27, 33	120	21∼28	4	2, 3, 4, 5
4, 14, 22	60	35, 37	140	29∼34	5	2, 3, 4, 5, 6
5, 15, 23, 29	70	10, 20, 28, 34	150	35, 36	4	5, 4, 5, 3
6, 16, 24, 30	80	36, 38	180	37, 38	5	4, 5, 3, 6, 3

评价多目标优化算法结果的指标很多,本文采用DI_R和C对算法的性能进行评价. 选择competitive memetic algorithm(CMA)[19]、多目标禁忌搜索(MOTS)[7]和非劣排序遗传算法-II(NSGA-II)[12]作为对比算法. 对于CMA,将本文的编码方式运用到该算法,同时将CMA中总延迟时间最大的工厂改为拖后工件数最大的工厂. MOTS在去掉维修相关的处理、增加工厂分配串、等概率选择3个邻域结构之一作用于工厂分配串后可用来解决DTHFSP,其中调度串和工厂分配串的邻域搜索依次进行. NSGA-II直接运用本文的编码和解码过程,等概率执行调度串和工厂分配串的两点交叉,变异时以相等的概率选择工厂分配串和调度串进行邻域搜索,对所选中的串等概率执行3个邻域结构中的一个.

为了验证改进SFLA模因组搜索过程的有效性,构建SFLA1. 该算法与改进SFLA的区别在于: 整个搜索过程只有一个阶段,采用改进SFLA早期阶段的重构和搜索方式,并将改进SFLA针对最优模因组的局部搜索方式加到SFLA1每个模因组的搜索过程中,同时加入对全局最优个体的局部搜索.

3.2　结果分析

经过大量仿真实验,得出5种算法的最佳参数设置如下:

改进SFLA: 种群规模为64,模因组个数为8,记忆容量为20,早期搜索模因组内搜索次数$\mu = 80$,早期第1类模因组组内进化次数为$\mu_1 = 120$,第2类模因组组内进化次数为$\mu_2 = 20$,第3类模因组组内进化次数为$\mu_3 = 60$,当目标函数估计次数达到2×10^4时,早期搜索过程结束,$\beta = 15, \theta = 0.5$,终止条件为最大目标函数估计次数$\max_\mathrm{it} = 10^5$.

SFLA1: 种群规模、模因组数量和终止条件与改进SFLA相同,模因组内搜索次数$\mu = 80$.

CMA: 终止条件与改进SFLA一致,其余参数均与原CMA相同.

MOTS: 邻域解的数量为300,终止条件与改进SFLA一致.

NSGA-II: 种群规模为100,交叉概率为0.8,变异概率为0.1,最大代数为1 000.

5种算法关于每个实例各独立运行20次,计算时间如表2所示. DI_R指标和C指标的计算结果如表3和表4所示. 配对样本均值t检验的结果如表5所示. 由表3和表4可以看出,改进SFLA的性能明显优于SFLA1,表明模因组分类并针对不同模因组采用不同搜索方式的策略合理有效,同样改进SFLA的性能也优于其3个对比算法. 表2从统计意义上得出的结果与以上分析一致. 如上所述,改进SFLA的搜索优势来源于模因组分类和不同类模因组搜索策略各异,因此,改进SFLA在解决具有SDST的DTHFSP方面具有较强的能力和优势.

表2　4种算法的计算时间

instance	SFLA	CMA	MOTS	NSGA-II	instance	SFLA	CMA	MOTS	NSGA-II	instance	SFLA	CMA	MOTS	NSGA-II
1	0.976	0.919	1.010	1.551	14	2.671	2.361	2.283	3.020	27	8.562	7.981	7.025	8.600
2	1.416	1.274	1.329	1.950	15	3.341	2.966	2.858	3.628	28	12.458	11.806	10.689	12.592
3	1.975	1.761	1.745	2.381	16	4.201	3.682	3.394	4.271	29	3.486	3.091	2.847	4.075
4	2.462	2.201	2.280	2.987	17	4.948	4.799	4.304	5.076	30	4.241	3.901	3.571	4.804
5	3.159	2.771	2.855	3.610	18	5.937	5.912	5.370	6.100	31	5.336	4.886	4.253	6.000
6	3.804	3.613	3.513	4.343	19	7.855	7.816	7.051	7.899	32	6.021	5.852	4.927	6.737
7	4.872	4.409	3.904	4.721	20	12.106	11.143	10.665	12.062	33	8.396	8.419	7.018	8.969
8	5.852	5.348	5.026	6.173	21	2.209	1.886	1.987	2.844	34	12.447	11.970	11.465	12.784
9	7.886	7.332	6.757	7.674	22	2.664	2.566	2.344	3.225	35	11.556	10.838	10.417	11.393
10	11.912	10.935	10.774	11.207	23	3.558	3.052	2.889	3.947	36	17.947	16.427	14.424	17.085
11	1.065	0.964	1.021	1.690	24	4.131	3.721	3.691	4.841	37	11.268	10.421	9.378	11.788
12	1.525	1.370	1.480	2.087	25	5.305	4.683	4.150	5.439	38	18.130	15.851	14.950	17.682
13	2.056	1.844	1.783	2.552	26	6.146	5.760	5.206	6.504					

表3 5种算法关于指标DI_R的计算结果

instance	SFLA	SFLA1	CMA	MOTS	NSGA-II	instance	SFLA	SFLA1	CMA	MOTS	NSGA-II
1	24.131	13.934	14.056	22.336	15.346	20	1.201	13.757	11.915	32.252	19.295
2	10.992	20.671	9.058	13.009	19.207	21	5.115	9.268	11.263	2.193	18.722
3	6.663	10.570	9.840	12.223	12.410	22	13.056	17.927	15.558	17.983	27.780
4	2.823	11.242	3.433	7.918	16.569	23	14.943	22.164	14.560	24.376	29.331
5	8.564	16.723	16.110	16.816	14.272	24	12.180	18.168	17.552	14.955	23.872
6	5.477	19.459	16.737	15.539	24.874	25	2.572	7.653	11.058	8.890	19.245
7	5.919	13.617	13.707	12.331	8.023	26	0.000	15.527	9.755	14.319	23.029
8	1.864	16.465	11.593	14.877	29.632	27	4.201	16.592	11.540	17.155	27.682
9	6.393	11.277	7.376	28.712	16.901	28	10.511	11.016	14.388	24.167	22.485
10	1.536	5.791	13.414	34.794	14.517	29	17.513	11.797	13.429	0.000	34.198
11	18.524	19.755	13.715	0.000	17.738	30	14.931	13.957	1.933	8.397	27.799
12	18.785	18.333	14.123	11.950	13.447	31	4.384	14.688	7.322	7.213	39.718
13	9.019	14.993	6.789	10.666	22.951	32	6.869	9.069	10.634	8.450	23.926
14	11.031	17.273	11.549	5.667	27.296	33	1.971	3.150	9.571	12.243	25.656
15	3.244	11.040	4.355	6.406	14.284	34	0.000	10.661	8.232	11.120	22.008
16	5.989	17.072	7.777	14.974	26.024	35	12.318	17.050	13.672	20.238	26.078
17	4.750	11.394	10.602	7.631	13.468	36	9.115	20.340	14.617	27.798	46.132
18	2.494	5.311	9.309	14.464	22.390	37	2.642	10.811	5.426	14.462	31.420
19	5.577	13.565	13.680	25.982	27.379	38	4.373	9.673	7.392	19.273	31.603

表4 5种算法关于指标C的计算结果

instance	$C(S,S_1)$	$C(S_1,S)$	$C(S,C)$	$C(C,S)$	$C(S,M)$	$C(M,S)$	$C(S,N)$	$C(N,S)$
1	0.000	0.857	0.000	1.000	0.000	0.857	0.000	0.714
2	0.750	0.167	0.500	0.000	0.500	0.333	1.000	0.000
3	0.833	0.000	0.600	0.200	0.750	0.200	1.000	0.000
4	0.600	0.143	0.700	0.143	0.800	0.143	1.000	0.000
5	1.000	0.000	1.000	0.000	0.846	0.000	1.000	0.000
6	1.000	0.000	0.333	0.200	1.000	0.000	1.000	0.000
7	0.571	0.000	0.429	0.000	0.929	0.000	0.750	0.000
8	1.000	0.000	0.625	0.000	0.857	0.000	1.000	0.000
9	1.000	0.000	0.700	0.500	1.000	0.000	0.750	0.000
10	0.500	0.125	1.000	0.000	1.000	0.000	1.000	0.000
11	0.625	0.000	0.250	0.000	0.000	1.000	0.500	0.000
12	0.286	0.429	0.000	1.000	0.222	0.571	0.500	0.429
13	0.750	0.000	0.000	0.750	0.500	0.250	1.000	0.000
14	0.889	0.000	0.500	0.250	0.200	0.250	0.667	0.000
15	1.000	0.000	0.800	0.000	1.000	0.000	0.800	0.000
16	0.800	0.000	0.889	0.000	1.000	0.000	1.000	0.000
17	0.857	0.143	0.583	0.286	0.733	0.143	0.333	0.286
18	0.571	0.429	0.333	0.143	1.000	0.000	1.000	0.000
19	1.000	0.000	0.909	0.000	1.000	0.000	1.000	0.000
20	1.000	0.000	0.875	0.000	1.000	0.000	1.000	0.000
21	0.800	0.000	0.750	0.000	0.500	0.500	1.000	0.000
22	0.000	0.800	0.143	0.200	0.000	1.000	1.000	0.000
23	0.667	0.000	0.000	0.250	0.500	0.250	1.000	0.000
24	0.750	0.000	0.750	0.000	0.818	0.000	0.800	0.000
25	0.750	0.250	1.000	0.000	1.000	0.000	1.000	0.000
26	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000
27	1.000	0.000	0.917	0.000	1.000	0.000	1.000	0.000
28	0.750	0.000	0.750	0.000	1.000	0.000	1.000	0.000
29	0.000	0.500	0.167	0.500	0.000	1.000	1.000	0.000
30	0.714	0.000	0.222	0.500	0.167	0.750	1.000	0.000
31	1.000	0.000	0.625	0.333	0.833	0.333	1.000	0.000
32	0.600	0.429	0.917	0.000	0.667	0.286	1.000	0.000
33	0.667	0.200	1.000	0.000	0.875	0.000	1.000	0.000
34	1.000	0.000	1.000	0.000	1.000	0.000	1.000	0.000
35	1.000	0.000	0.900	0.000	0.909	0.000	1.000	0.000
36	1.000	0.000	0.615	0.000	1.000	0.000	1.000	0.000
37	0.800	0.000	0.778	0.000	1.000	0.000	1.000	0.000
38	0.750	0.125	0.769	0.000	1.000	0.000	1.000	0.000

注：S表示改进SFLA, S_1表示SFLA1, C表示CMA, M表示MOTS, N表示NSGA-II

表5　配对样本t检验

t检验	p-value(DI_R)	p-value(C)
t-test(SFLA, SFLA1)	0.000	0.000
t-test(SFLA, CMA)	0.001	0.000
t-test(SFLA, MOTS)	0.000	0.000
t-test(SFLA, NSGA-II)	0.000	0.000

4　结　论

本文针对具有SDST的双目标DTHFSP,提出了一种改进SFLA以最小化makespan和拖后工件数. 采用启发式方法初始化,提高了初始种群的质量,对模因组进行分类,每一类模因组的搜索过程各异,并分配给最好模因组更多的搜索次数,减少最差模因组的搜索次数,从而避免无效搜索引起的资源浪费,保持种群的多样性,而最好模因组不参与种群划分,可维持其良好的解的组成. 计算实验结果表明,改进SFLA在解决DTHSP方面具有较强的搜索优势.

目前,DHFSP的研究才刚刚起步,未来将进一步深入研究考虑各种实际加工约束的多目标DHFSP,并探讨人工蜂群算法和SFLA等算法在解决该类问题方面的搜索能力和优势.

参考文献(References)

[1] Choong F, Phon-Amnuaisuk S, Alias M Y. Metaheuristic methods in hybrid flow shop scheduling problem[J]. Expert Systems with Applications, 2011, 38(9): 10787-10793.

[2] Engin O, Ceran G, Yilmaz M K. An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems[J]. Applied Soft Computing, 2011, 11(3): 3056-3065.

[3] Liao C J, Tjandradjaja E, Chung T P. An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem[J]. Applied Soft Computing, 2012, 12(6): 1755-1764.

[4] Li J Q, Pan Q K, Wang F T. A hybrid variable neighborhood search for solving the hybrid flow shop scheduling problem[J]. Applied Soft Computing, 2014, 24: 63-77.

[5] Marichelvam M K, Prabaharan T, Yang X S. Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan[J]. Applied Soft Computing, 2014, 19(1): 93-101.

[6] Fattahi P, Hosseini S M H, Jolai F. A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations[J]. Applied Mathematical Modelling, 2014, 38(1): 119-134.

[7] Wang S J, Liu M. Two-stage hybrid flow shop scheduling with preventive maintenance using multi-objective tabu search method[J]. International Journal of Production Research, 2014, 52(5): 1495-1508.

[8] Jun S, Park J. A hybrid genetic algorithm for the hybrid flow shop scheduling problem with nighttime work and simultaneous work constraints: A case study from the transformer industry[J]. Expert Systems with Applications, 2015, 42(15/16): 6196-6204.

[9] Jiang S L, Liu M, Hao J H, et al. A bi-layer optimization approach for a hybrid flow shop scheduling problem involving controllable processing times in the steelmaking industry[J]. Computers & Industrial Engineering, 2015, 87: 518-531.

[10] 王圣尧, 王凌, 许烨, 等. 求解混合流水车间调度问题的分布估计算法[J]. 自动化学报, 2012, 38(3): 437-443.
(Wang S Y, Wang L, Xu Y, et al. An estimation of distribution algorithm for solving hybrid flow-shop scheduling problem[J]. Acta Automatica Sinica, 2012, 38(3): 437-443.)

[11] 田云娜, 李冬妮, 郑丹, 等. 一种基于时间窗的多阶段混合流水车间调度方法[J]. 机械工程学报, 2016, 52(16): 185-196.
(Tian Y N, Li D N, Zheng D, et al. A time window-based approach for multi-stage hybrid flow shop[J]. Journal of Mechanical Engineering, 2016, 52(16): 185-196.)

[12] Ying K C, Lin S W. Minimizing makespan for the distributed hybrid flowshop scheduling problem with multiprocessor tasks[J]. Expert Systems with Applications, 2018, 92: 132-141.

[13] 蔡劲草, 雷德明. 考虑准备时间的分布式两阶段混合流水车间调度[J]. 计算机集成制造系统, 2020, 26(8): 2170-2179.
(Cai J C, Lei D M. Distributed two-stage hybrid flow shop scheduling with setup times[J]. Computer Integrated Manufacturing Systems, 2020, 26(8): 2170-2179.)

[14] Lin S W, Ying K C. Minimizing makespan for solving the distributed no-wait flowshop scheduling problem[J]. Computers & Industrial Engineering, 2016, 99: 202-209.

[15] Ying K C, Lin S W, Cheng C Y, et al. Iterated reference greedy algorithm for solving distributed noidle permutation flowshop scheduling problems[J]. Computers and Industrial Engineering, 2017, 110: 413-423.

[16] Ruiz R, Pan Q K, Naderi B. Iterated Greedy methods for the distributed permutation flowshop scheduling problem[J]. Omega, 2019, 83: 213-222.

[17] Hatami S, Ruiz R, Andrés-Romano C. Heuristics and metaheuristics for the distributed assembly permutation flowshop scheduling problem with sequence dependent setup times[J]. International Journal of Production Economics, 2015, 169: 76-88.

[18] Gonzalez-Neira E M, Ferone D, Hatami S, et al. A biased-randomized simheuristic for the distributed

assembly permutation flowshop problem with stochastic processing times[J]. Simulation Modelling Practice and Theory, 2017, 79: 23-36.

[19] Deng J, Wang L. A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem[J]. Swarm and Evolutionary Computation, 2017, 32: 121-131.

[20] Rifai A P, Nguyen H T E M, Dawal S Z M. Multi-objective adaptive large neighborhood search for distributed reentrant permutation flow shop scheduling[J]. Applied Soft Computing, 2016, 40: 42-57.

[21] Eusuff M M, Lansey K E. Optimization of water distribution network design using the shuffled frog leaping algorithm[J]. Journal of Water Resources Planning and Management, 2003, 129(3): 210-225.

[22] Eusuff M, Lansey K, Pasha F. Shuffled frog-leaping algorithm: A memetic meta-heuristic for discrete optimization[J]. Engineering Optimization, 2006, 38(2): 129-154.

[23] Rahimi-Vahed A, Mirzaei A H. Solving a bi-criteria permutation flow-shop problem using shuffled frog leaping algorithm[J]. Soft Computing, 2008, 12(5): 435-452.

[24] Chung G, Lansey K. Application of the shuffled frog leaping algorithm for the optimization of a general large-scale water supply system[J]. Water Resources Management, 2009, 23(4): 797-823.

[25] Rahimi-Vahed A, Mirzaei A H. A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem[J]. Computers & Industrial Engineering, 2007, 53(4): 642-666.

[26] Rahimi-Vahed A, Dangchi M, Rafiei H, et al. A novel hybrid multi-objective shuffled frog-leaping algorithm for a bi-criteria permutation flow shop scheduling problem[J]. International Journal of Advanced Manufacturing Technology, 2009, 41(11/12): 1227-1239.

[27] Xu Y, Wang L, Liu M, et al. An effective shuffled frog-leaping algorithm for hybrid flow-shop scheduling with multiprocessor tasks[J]. International Journal of Advanced Manufacturing Technology, 2013, 68(5/6/7/8): 1529-1537.

[28] Xu Y, Wang L, Wang S Y, et al. An effective shuffled frog-leaping algorithm for solving the hybrid flow-shop scheduling problem with identical parallel machines[J]. Engineering Optimization, 2013, 45(12): 1409-1430.

[29] 艾子义, 雷德明. 基于新型蛙跳算法的低碳柔性作业车间调度[J]. 控制理论与应用, 2017, 34(10): 1361-1368.
(Ai Z Y, Lei D M. A novel shuffled frog leaping algorithm for low carbon flexible job shop scheduling[J]. Control Theory and Applications, 2017, 34(10): 1361-1368.)

[30] Lei D M, Guo X P. A shuffled frog-leaping algorithm for hybrid flow shop scheduling with two agents[J]. Expert Systems with Applications, 2015, 42(23): 9333-9339.

[31] Lei D M, Zheng Y L, Guo X P. A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption[J]. International Journal of Production Research, 2017, 55(11): 3126-3140.

作者简介

雷德明(1968−), 男, 教授, 博士生导师, 从事智能系统优化与控制等研究, E-mail: deminglei11@163.com;

王甜(1993−), 女, 硕士生, 从事制造系统智能优化与调度的研究, E-mail: wangtianzdh@126.com.

(责任编辑：郑晓蕾)

《控制与决策》入选“2020中国国际影响力优秀期刊”“2020年度中国高校百佳科技期刊”

本刊讯： 2020年12月17日,由中国期刊协会、中国科学技术期刊编辑学会、中国高校科技期刊研究会、全国高等学校文科学报研究会、《中国学术期刊(光盘版)》电子杂志社有限公司共同主办的“2020中国学术期刊未来论坛”以线上会议形式隆重召开. 会上发布了“TOP10%中国国际影响力品牌学术期刊”榜单,《控制与决策》入选“2020中国国际影响力优秀期刊”. 大会同时发布了《世界期刊影响力指数(WJCI)报告(2020科技版)中国期刊名单》,《控制与决策》在“控制科学与技术”学科中(WJCI分区为Q1区)位列中国期刊的第一位.

另悉,《控制与决策》被中国高校科技期刊研究会评为“2020年度中国高校百佳科技期刊”.