

# 控制与决策

Control and Decision

## 超启发式交叉熵算法求解模糊分布式流水线绿色调度问题

钱斌, 余明哲, 胡蓉, 郭宁, 向凤红

引用本文:

钱斌, 余明哲, 胡蓉, 等. 超启发式交叉熵算法求解模糊分布式流水线绿色调度问题[J]. *控制与决策*, 2021, 36(6): 1387–1396.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.1681>

---

### 您可能感兴趣的其他文章

Articles you may be interested in

#### [基于正态云模型的状态转移算法求解多目标柔性作业车间调度问题](#)

State transition algorithm based on normal cloud model for solving multi-objective flexible job shop scheduling problem

*控制与决策*. 2021, 36(5): 1181–1190 <https://doi.org/10.13195/j.kzyjc.2019.1233>

#### [带不相关并行机和有限缓冲MHFS调度的混合启发式算法](#)

Hybrid heuristic algorithm for multi-stage hybrid flow shop scheduling with unrelated parallel machines and finite buffers

*控制与决策*. 2021, 36(3): 565–576 <https://doi.org/10.13195/j.kzyjc.2019.0835>

#### [基于改进NSGA-II算法求解多目标资源受限项目调度问题](#)

An improved NSGA-II algorithm for multi-objective resource-constrained project scheduling problem

*控制与决策*. 2021, 36(3): 669–676 <https://doi.org/10.13195/j.kzyjc.2019.0906>

#### [基于多班教学优化的多目标分布式混合流水车间调度](#)

Multi-class teaching-learning-based optimization for multi-objective distributed hybrid flow shop scheduling

*控制与决策*. 2021, 36(2): 303–313 <https://doi.org/10.13195/j.kzyjc.2020.0549>

#### [基于改进蛙跳算法的分布式两阶段混合流水车间调度](#)

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

*控制与决策*. 2021, 36(1): 241–248 <https://doi.org/10.13195/j.kzyjc.2019.0472>

# 超启发式交叉熵算法求解模糊分布式流水线绿色调度问题

钱斌<sup>1,2†</sup>, 余明哲<sup>1,2</sup>, 胡蓉<sup>1,2</sup>, 郭宁<sup>1</sup>, 向凤红<sup>1</sup>

- (1. 昆明理工大学 信息工程与自动化学院, 昆明 650500;
2. 昆明理工大学 云南省人工智能重点实验室, 昆明 650500)

**摘要:** 针对实际生产过程中普遍存在的加工时间不确定性, 采用模糊数表示工件的加工时间, 以同时最小化模糊最大完工时间和模糊总能耗为优化目标, 建立模糊分布式流水线绿色调度问题 (green distributed permutation flow-shop scheduling problem with fuzzy processing time, GDPFSP\_FPT) 的模型, 进而提出一种超启发式交叉熵算法 (hyper-heuristic cross-entropy algorithm, HHCE) 进行求解. 首先, HHCE 采用一种新颖的三角模糊数排序准则合理计算个体的目标函数值, 可在算法搜索过程中较准确发现优质解区域; 其次, HHCE 在高层利用基于贡献率的评价方法确定 8 种特定邻域操作所构成的各排列的优劣, 同时采用交叉熵 (cross-entropy, CE) 方法学习较优排列的信息并生成新排列, 进而在低层把高层生成的每个排列作为一种启发式算法, 对低层相应个体执行一系列邻域操作, 以实现问题解空间较多不同区域的搜索; 然后, HHCE 将基于非关键路径的节能策略用于对低层每代种群中的较优个体执行局部搜索, 从而进一步提高算法获取低能耗非劣个体或解的能力; 最后, 仿真实验与算法对比表明, HHCE 可有效求解 GDPFSP\_FPT.

**关键词:** 分布式流水线; 模糊加工时间; 绿色调度; 多目标优化; 超启发式算法

中图分类号: TP273 文献标志码: A

DOI: 10.13195/j.kzyjc.2019.1681

开放科学 (资源服务) 标识码 (OSID):



引用格式: 钱斌, 余明哲, 胡蓉, 等. 超启发式交叉熵算法求解模糊分布式流水线绿色调度问题 [J]. 控制与决策, 2021, 36(6): 1387-1396.

## Hyper-heuristic cross-entropy algorithm for green distributed permutation flow-shop scheduling problem with fuzzy processing time

QIAN Bin<sup>1,2†</sup>, SHE Ming-zhe<sup>1,2</sup>, HU Rong<sup>1,2</sup>, GUO Ning<sup>1</sup>, XIANG Feng-hong<sup>1</sup>

- (1. College of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China;
2. Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650500, China)

**Abstract:** For dealing with the processing time uncertainty widely existing in the real-world production process, this paper uses fuzzy number to represent each job's processing time, and establishes a model of the green distributed permutation flow-shop scheduling problem with fuzzy processing time (GDPFSP\_FPT), whose optimization objectives are the fuzzy maximum completion time and the fuzzy total energy consumption. The, a hyper-heuristic cross-entropy algorithm (HHCE) is proposed for solving the GDPFSP\_FPT. Firstly, the HHCE algorithm adopts a novel ranking rule of triangular fuzzy number to reasonably calculate the objective function values of individuals, which is helpful in finding the promising regions more accurately during the search process. Secondly, in the upper layer, the HHCE algorithm utilizes an evaluation method based on the contribution rate to estimate the permutations constructed by eight special neighbor operations, and also uses the cross-entropy (CE) method to learn the information of better permutations and generate new permutations. Then, for searching more different regions in solution space, the algorithm uses each permutation generated in the upper layer as a heuristic to perform a series of neighbor operations on the corresponding individuals in the lower layer. Thirdly, in order to enhance its ability of obtaining the non-dominated individuals or solutions with low energy consumption, the algorithm utilizes an energy-saving strategy based on non-critical path to perform local search on better individuals of each generation. Finally, simulations and comparisons demonstrate that the HHCE algorithm can effectively solve the GDPFSP\_FPT.

**Keywords:** distributed permutation flow-shop; fuzzy processing time; green scheduling; multi-objective optimization; hyper-heuristic algorithm

收稿日期: 2019-11-29; 修回日期: 2020-01-13.

基金项目: 国家自然科学基金项目 (51665025, 61963022).

责任编辑: 刘民.

†通讯作者. E-mail: bin.qian@vip.163.com.

## 0 引言

在经济全球化的背景下,制造活动开始从集中式向分布式进行转变.分布式制造有效减少了规模效应,工厂处理能力被分散到各地,因此这种制造方式能够快速重新定义需求,制定更为符合实际的生产计划.同时,生产过程中存在多种不确定因素<sup>[1]</sup>,这类因素导致在确定条件假设下得到的优质调度方案在实际生产过程中往往不再具有显著的指导意义,所以利用模糊理论中的模糊数表征不确定的加工时间,有助于更为客观地描述实际生产过程.另外,分布式制造在获得经济效益的同时也对环境造成了一定程度的影响.让绿色理念推动传统企业转型,让绿色制造助力制造强国建设已成为我国制造业发展的新主题.因此,研究模糊分布式流水线绿色调度问题(green distributed permutation flow-shop scheduling problem with fuzzy processing time, GDPFSP\_FPT)具有现实的经济价值和深远的社会意义.在计算复杂度上,分布式流水线调度问题(distributed permutation flow-shop scheduling problem, DPFSP)已被证明具有非确定多项式难(non-deterministic polynomial hard, NP-hard)的属性,而该问题又归约为GDPFSP\_FPT,故GDPFSP\_FPT属于NP-hard问题.因此,研究面向GDPFSP\_FPT问题及其求解算法具有重要的理论价值和实际意义.

近年来,分布式流水线调度问题得到了广泛的关注与研究,文献[2]提出了分布式流水线调度问题,并以最小化最大完工时间为目标,提出了6个混合整数线性规划模型和2个分配规则,进而基于规则提出了14种启发式算法和变邻域下降算法(variable neighborhood descent method, VND)进行求解;文献[3-4]针对优化目标为最小化最大完工时间的分布式流水线调度问题,分别设计了有界搜索迭代贪婪算法(bounded-search iterated greedy algorithm, BSIG)和化学反应优化(chemical reaction optimization, CRO)算法进行求解;文献[5]针对优化目标为最小化最大完工时间的分布式阻塞流水线调度问题,设计了一种离散差分进化(discrete differential evolution, DDE)算法进行求解;文献[6]针对优化目标为最小化最大完工时间的分布式异构流水线调度问题,设计了一种基于概率模型的文化基因算法(probability model-based memetic algorithm, PMMA)进行求解.

实际生产过程大多存在不确定性.受设备状况、工人熟练程度等影响,工件或工序加工时间往往只能确定一个大致范围.因此,采用模糊数表示加工时间有益于获得鲁棒、可行的调度方案.目前,已有学者对

模糊加工时间的生产调度问题进行研究.文献[7]针对优化目标为最小化模糊最大完工时间和最大化交货期满意度的模糊开放车间调度问题,设计了一种多目标粒子群优化算法(multi-objective particle swarm optimization algorithm, MOPSO)进行求解;文献[8]针对优化目标为同时最小化模糊最大完工时间和模糊机器最大负荷的模糊柔性作业车间调度问题,设计了一种改进人工蜂群算法(improved artificial bee colony algorithm, IABC)进行求解;文献[9]针对优化目标为同时最小化拖期惩罚与总能耗的带状态相关设置时间模糊流水车间调度问题,设计了一种经典遗传算法与启发式算法相结合的混合算法进行求解;文献[10]针对优化目标为最小化模糊最大完工时间的模糊柔性作业车间调度问题,设计了一种混合协同进化算法(hybrid cooperative coevolution algorithm, HCEA)进行求解.目前,对于模糊分布式流水线调度问题(distributed permutation flow-shop scheduling problem with fuzzy processing time, DPFSP\_FPT)尚未见相关研究报道.

与传统调度问题相比,绿色调度问题通过资源分配、操作排序和运作模式的合理优化,实现节能减排、降本增效,其求解难度更大,也更具工程意义和理论价值<sup>[11]</sup>.在绿色分布式流水线调度问题(green distributed permutation flow-shop scheduling problem, GDPFSP)研究方面,文献[12]针对优化目标为同时最小化最大完工时间和碳排放总量的分布式流水线车间调度问题,设计了一种竞争文化基因算法(competitive memetic algorithm, CMA)进行求解;文献[13]针对带有总延迟约束,优化目标为同时最小化最大完工时间和总能耗的分布式流水线调度问题,设计了一种多目标头脑风暴优化算法(multi-objective brain storm optimization algorithm, MOBSO)进行求解.根据文献调研,绿色分布式流水线调度问题的研究仍十分有限,同时没有模糊分布式流水线绿色调度问题的相关研究.

超启发式算法(hyper-heuristic algorithm, HH)是一类新型智能优化算法.该算法通过某种高层策略(high-level strategy, HLS)确定或生成低层启发式算法(low-level heuristics, LLH),以实现问题解空间的搜索.文献[14]提出了超启发式算法的概念并将其用于求解调度问题;近年来,在车间调度方面,文献[15]针对优化目标最小化期望最大完工时间的随机作业车间调度问题,提出了一种自适应超启发式遗传算法(adaptive hyper-heuristics genetic algorithm, AHHGA),高层利用带随机交换变异操作的遗传算法,对不同故障场景下的4种启发式规则的排列进行优化,低层

按照高层所得排列依次执行规则,继而生成新解并进行搜索;文献[16]针对优化目标为最小化最大完工时间的分布式流水线装配调度问题,提出了一种回溯搜索超启发式算法(backtracking search hyper-heuristic algorithm, BSHH),高层利用回溯搜索算法对由10个低层启发式操作组成的操作序列进行优化,低层按照高层所得排列进行搜索.超启发式算法已被成功用于求解多种车间调度问题,但尚未用于求解分布式流水线调度问题.

交叉熵(cross-entropy, CE)方法是由文献[17]提出的一种小概率事件估计方法.该方法利用稀有事件信息更新相应的概率模型参数,并使用有效采样方法对模型采样以生成新种群,进而可合理引导搜索方向.因此,已有学者将CE方法应用到组合优化问题的求解中.譬如,文献[18]针对优化目标为最小化最大完工时间的零等待作业车间调度问题,提出了一种混合交叉熵遗传算法(cross entropy-genetic algorithm, CEGA)进行求解;文献[19]针对加工时间不确定的炼钢连铸生产调度问题,提出了一种串级交叉熵算法(cascade cross-entropy algorithm, CCEA)进行求解.

本文研究GDPFSP\_FPT的建模与求解.在建模方面,针对实际生产中广泛存在的加工时间不确定性,采用三角模糊数表征加工时间,建立以同时最小化模糊最大完工时间和模糊总能耗为优化目标的GDPFSP\_FPT模型.在求解方面,考虑到GDPFSP\_FPT这类问题的解空间复杂且可变,常规智能算法难以实现有效搜索,故设计一种融合HH与CE方法的超启发式交叉熵算法(hyper-heuristic cross-entropy algorithm, HHCE)进行求解.HHCE由高层策略域与低层问题域组成,其中高层优化8种特定邻域操作所构成的排列,低层用高层确定的操作排列作为启发式算法对问题解空间进行搜索.具体而言:在HHCE的高层,根据多目标问题解的特点,设计基于贡献率的评价方法确定8种特定邻域操作所构成的不同排列的优劣,并采用交叉熵概率模型学习和积累部分较优排列的信息,进而采样该模型以生成新的操作排列;在HHCE的低层,把高层生成的每个排列作为一种启发式算法作用于一定数量的低层个体,以实现了解空间较多不同区域的搜索.此外,设计基于非关键路径的节能策略,在保持关键路径上各操作加工速度不变的前提下,调整非关键路径上部分操作的加工速度以降低总能耗,从而进一步提高算法获取低能耗非劣个体或解的能力.最后,通过仿真实验和算法比较验证所提算法的有效性.

## 1 模糊分布式流水线绿色调度问题

### 1.1 符号定义

关于本文涉及的数学符号及定义如下:

$f$ :工厂序列,  $f = \{1, 2, \dots, F\}$ ;

$j$ :机器序列,  $j = \{1, 2, \dots, M\}$ ;

$k$ :工件序列,  $k = \{1, 2, \dots, N\}$ ;

$V_R$ :机器加工速度序列,  $V_R = \{v_{r_1}, v_{r_2}, \dots, v_{r_s}\}$ ;

popsize:问题域种群容量;

$\varepsilon$ :机器空载时的单位能耗;

$o_{k,j}$ :工件 $k$ 在机器 $j$ 上的操作;

$\pi$ :工件加工序列,  $\pi = \{\pi_1, \pi_2, \dots, \pi_N\}$ ;

$\pi^f$ :工厂 $f$ 的工件加工序列,  $\pi^f = \{\pi_1^f, \pi_2^f, \dots, \pi_{N_f}^f\}$ ;

$V_N$ :工件加工速度对照表,

$$V_N = \begin{bmatrix} v_{1,1} & \cdots & v_{k,1} & \cdots & v_{N,1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ v_{1,j} & \cdots & v_{k,j} & \cdots & v_{N,j} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ v_{1,M} & \cdots & v_{k,M} & \cdots & v_{N,M} \end{bmatrix}, v_{k,j} \in V_R;$$

$\tilde{p}_{k,j}$ :工件 $k$ 在机器 $j$ 上的标准加工时间;

$\tilde{t}_{k,j}$ :工件 $k$ 在机器 $j$ 上的实际加工时间;

$\tilde{C}_{k,j}$ :工件 $k$ 在机器 $j$ 上的完工时间;

$\tilde{C}_f$ :工厂 $f$ 的完工时间;

$\widetilde{LEC}_f$ :工厂 $f$ 的总带载能耗;

$\widetilde{NLEC}_f$ :工厂 $f$ 的总空载能耗;

$\widetilde{TEC}_f$ :工厂 $f$ 的总能耗;

$\widetilde{TEC}$ :模糊总能耗;

$\tilde{C}_{\max}$ :模糊最大完工时间.

### 1.2 问题模型

GDPFSP\_FPT可描述为:将 $N$ 个工件分配给 $F$ 个工厂,每个工厂有一条 $M$ 台机器的流水线,每台机器的加工速度 $v$ 均有 $s$ 个档位可供调节.每个工件可在任意一个工厂依次完成 $M$ 道加工操作.每台机器在同一时刻只能加工一个工件.每个工件在同一时间只能由一台机器进行加工.每个操作 $o_{k,j}$ 在加工过程中均不可中断,且都具有相应的标准加工时间 $\tilde{p}_{k,j}$ .每台机器的加工速度一旦选定,在操作 $o_{k,j}$ 结束前不可更改,且此时操作对应的实际加工时间为 $\tilde{t}_{k,j}$ .

GDPFSP\_FPT的目标函数为模糊最大完工时间 $\tilde{C}_{\max}$ 和模糊总能耗 $\widetilde{TEC}$ .基于问题描述,建立如下模型:

$$\tilde{t}_{k,j} = \frac{\tilde{p}_{k,j}}{\tilde{v}_{k,j}}, k = 1, 2, \dots, N, j = 1, 2, \dots, M; \quad (1)$$

$$\tilde{C}_{\pi_1^f, 1} = \tilde{t}_{\pi_1^f, 1}, f = 1, 2, \dots, F; \quad (2)$$

$$\begin{aligned} \tilde{C}_{\pi_i^f,1} &= \tilde{C}_{\pi_{i-1}^f,1} + \tilde{t}_{\pi_i^f,1}, \\ i &= 2, 3, \dots, N_f, f = 1, 2, \dots, F; \end{aligned} \quad (3)$$

$$\begin{aligned} \tilde{C}_{\pi_1^f,j} &= \tilde{C}_{\pi_1^f,j-1} + \tilde{t}_{\pi_1^f,j}, \\ j &= 2, 3, \dots, M, f = 1, 2, \dots, F; \end{aligned} \quad (4)$$

$$\begin{aligned} \tilde{C}_{\pi_i^f,j} &= \tilde{C}_{\pi_{i-1}^f,j} \vee \tilde{C}_{\pi_i^f,j-1} + \tilde{t}_{\pi_i^f,j}, \\ i &= 2, 3, \dots, N_f, j = 2, 3, \dots, M, f = 1, 2, \dots, F; \end{aligned} \quad (5)$$

$$\tilde{C}_f = \tilde{C}_{\pi_{N_f}^f,M}, f = 1, 2, \dots, F; \quad (6)$$

$$\tilde{C}_{\max} = \max[\tilde{C}_1, \tilde{C}_2, \dots, \tilde{C}_F]; \quad (7)$$

$$\widetilde{\text{LEC}}_f = \sum_{j=1}^M \sum_{i=1}^{N_f} 4 \cdot (v_{\pi_i^f,j})^2 \cdot \tilde{t}_{\pi_i^f,j}, f = 1, 2, \dots, F; \quad (8)$$

$$\begin{aligned} \widetilde{\text{NLEC}}_f &= \sum_{j=1}^M \sum_{i=2}^{N_f} \varepsilon \cdot [(\tilde{C}_{\pi_i^f,j} - \tilde{C}_{\pi_{i-1}^f,j} - \tilde{t}_{\pi_i^f,j}) \vee \\ &(0, 0, 0)], f = 1, 2, \dots, F; \end{aligned} \quad (9)$$

$$\widetilde{\text{TEC}}_f = \widetilde{\text{LEC}}_f + \widetilde{\text{NLEC}}_f, f = 1, 2, \dots, F; \quad (10)$$

$$\widetilde{\text{TEC}} = \sum_{f=1}^F \widetilde{\text{TEC}}_f. \quad (11)$$

1.3 模糊数及其运算

在计算模糊生产调度问题不同个体或解对应的目标函数值时,需对模糊数(即加工时间)进行加、减、乘、除、取大运算.同时,在评价不同解的优劣时,需进行比较运算.

根据模糊数学的有关定义和扩张原理,设  $\tilde{U} = (u_1, u_2, u_3)$  和  $\tilde{W} = (w_1, w_2, w_3)$  为两个三角模糊数,定义其加法运算如下:

$$\tilde{U} + \tilde{W} = (u_1 + w_1, u_2 + w_2, u_3 + w_3). \quad (12)$$

定义其减法运算如下:

$$\tilde{U} - \tilde{W} = (u_1 - w_1, u_2 - w_2, u_3 - w_3). \quad (13)$$

$\xi$  为任意实数,定义其乘法运算如下:

$$\xi \cdot \tilde{U} = (\xi \cdot u_1, \xi \cdot u_2, \xi \cdot u_3). \quad (14)$$

$\xi$  为任意非零实数,定义其除法运算如下:

$$\frac{\tilde{U}}{\xi} = \left( \frac{u_1}{\xi}, \frac{u_2}{\xi}, \frac{u_3}{\xi} \right). \quad (15)$$

取大、比较运算均由模糊数排序实现.最近,文献[20]在充分考虑取大操作的近似误差和模糊度基础上,提出了一种针对三角模糊数的模糊排序准则,并通过理论分析和实验比较验证所提准则优于文献[21]提出的且在近年调度文献中广泛使用的排序准则.因此,本文算法采用文献[20]的排序准则计算

问题解中每个工件在各机器上对应操作的模糊开始加工时间.这可较合理计算个体或解的目标函数值,从而有利于算法在搜索过程中较准确找到优质解区域.该排序准则具体定义如下.

记  $u_1u_2$  与  $w_1w_2, u_2u_3$  与  $w_2w_3$  分别为两组对应边;记  $u_1u_2$  与  $w_2w_3, u_2u_3$  与  $w_1w_2$  分别为两组非对应边.

step 1: 若两个 TFN 有对应边相交,记非对应边交点为  $(a, b)$ ,则定义  $L_0 = \left(\frac{1}{2} - a\right) \times (u_3 - w_1) + \frac{1}{2}(w_3 - u_1)$  用于判断.若  $L_0 > 0$ ,则  $\tilde{U} < \tilde{W}$ ,可得  $\tilde{U} \vee \tilde{W} = \tilde{W}$ ;若  $L_0 < 0$ ,则  $\tilde{U} > \tilde{W}$ ,可得  $\tilde{U} \vee \tilde{W} = \tilde{U}$ (上述内容均基于  $u_2 < w_2$  的假设,若  $u_2 > w_2$ ,则只需将式中的  $u$  与  $w$  调换位置即可);若  $L_0 = 0$  或没有对应边相交,则定义  $L_1(\tilde{U}) = \frac{u_1 + u_2 + u_3}{3}$  用于判断.若  $L_1(\tilde{U}) > L_1(\tilde{W})$ ,则  $\tilde{U} > \tilde{W}$ ,可得  $\tilde{U} \vee \tilde{W} = \tilde{U}$ ;否则,  $\tilde{U} < \tilde{W}$ ,可得  $\tilde{U} \vee \tilde{W} = \tilde{W}$ .

step 2: 若  $L_1(\tilde{U}) = L_1(\tilde{W})$ ,则定义  $L_2(\tilde{U}) = u_3 - u_1$  用于判断.若  $L_2(\tilde{U}) > L_2(\tilde{W})$ ,则  $\tilde{U} < \tilde{W}$ ,可得  $\tilde{U} \vee \tilde{W} = \tilde{W}$ ;否则,  $\tilde{U} > \tilde{W}$ ,可得  $\tilde{U} \vee \tilde{W} = \tilde{U}$ .

2 超启发式交叉熵算法

2.1 编码与解码

在 HHCE 中,首先在高层策略域利用 CE 方法对 8 种操作的组合顺序(每种顺序为一个高层种群个体)进行优化,获得高层策略域种群;然后在低层问题域将每个策略域个体作为独立的启发式算法,对低层相应个体进行变邻域局部搜索. HHCE 两层结构示意图如图 1 所示.

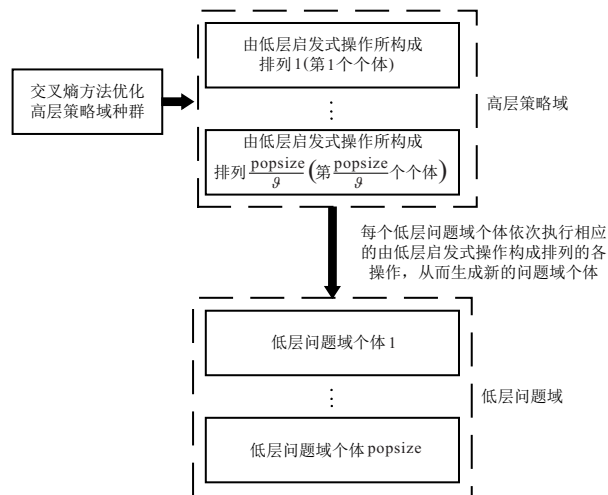


图 1 HHCE 的两层结构

对于高层策略域,种群中每个个体均由 8 种低层启发式操作构成,个体长度为 10,且同一个体中允许出现相同的低层启发式操作.由于一个高层策略域个体对应更新  $\vartheta$  个低层问题域个体,策略域种群容量为问题域种群容量的  $\frac{1}{\vartheta}$ ,即  $\frac{\text{popsize}}{\vartheta}$ .解码策略域个体

时,对低层问题域中的解,从左到右依次执行策略域个体每一位所代表的低层启发式操作,每执行完一个低层启发式操作,就对操作前后的两个解进行比较.若旧解支配新解,则有一定概率使用新解代替旧解,并继续执行剩余低层启发式操作;否则,使用新解代替旧解继续执行剩余低层启发式操作.高层策略域个体示意图如图2所示.



图2 高层策略域个体

对于低层问题域,每个个体就是原问题的一个解.问题域个体的编码由两部分组成:一部分为工件加工序列 $\pi$ ;另一部分为工件加工速度对照表 $V_N$ .其中,每台机器的加工速度均有6个档位可供调节( $V_R = \{1.0, 1.1, 1.2, 1.3, 1.4, 1.5\}$ ).例如,以 $N = 7, M = 2$ 的问题为例,低层问题域个体示意图如图

3所示,即第1个加工的工件为工件1,其对应加工速度为 $V_N$ 中第1列( $k = 1$ ),第2个加工的工件为工件3,其对应加工速度为 $V_N$ 中第3列( $k = 3$ ),依此类推.

1	3	4	7	2	6	5
$\pi_1$	$\pi_2$	$\pi_3$	$\pi_4$	$\pi_5$	$\pi_6$	$\pi_7$

(a) 工件加工序列  $\pi$

$j_1$	1.1	1.3	1	1.4	1.2	1	1.5
$j_2$	1.5	1.2	1	1.3	1	1.4	1.5
	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$

(b) 工件加工速度对照表  $V_N$

图3 工件加工序列

对于低层问题域个体采用文献[2]中所提的NR2规则进行解码.例如,对于 $F = 2, N = 7, M = 2, \pi$ 与 $V_N$ 如图3所示的问题.工件标准加工时间如表1所示.经由式(1)计算可得工件实际加工时间,解码后得到对应的甘特图如图4所示.

表1 工件标准加工时间

$j$	$k$						
	1	2	3	4	5	6	7
1	(5, 9, 13)	(2, 5, 11)	(11, 15, 16)	(6, 9, 12)	(6, 8, 10)	(2, 4, 7)	(3, 6, 9)
2	(3, 7, 10)	(7, 8, 11)	(7, 17, 15)	(5, 6, 8)	(4, 8, 11)	(1, 2, 3)	(2, 5, 11)

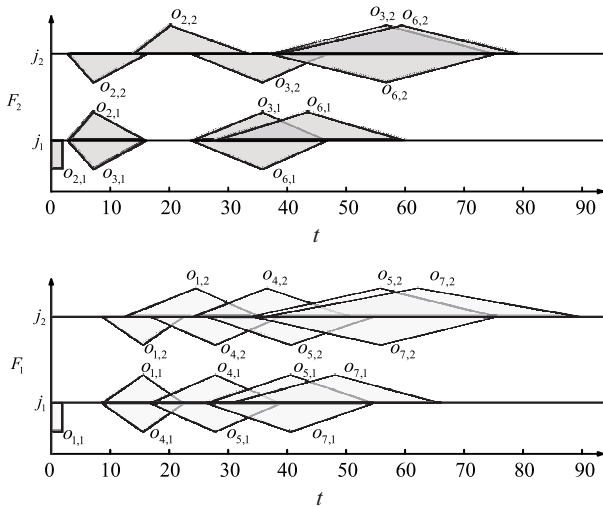


图4 低层问题域个体如图3所示时的甘特图

2.2 种群初始化

本文采用随机初始化方法分别生成高层策略域和低层问题域种群,避免解的分布过于集中,导致后续操作无法对解空间进行充分搜索.

2.3 低层启发式操作

对于组合优化问题,最小解(即最小化问题的最优解)在所有邻域结构下均为局部最小解,而接近最小解的优质解往往在多种邻域结构下均为局部最小解.采用单一的迭代邻域搜索容易较早达到并陷

入该邻域结构的局部最小解,而该解的质量大都一般.同时,GDPFSP\_FPT考虑多个并行的流水车间和各台机器速度可调,这使其解空间规模变得十分庞大和复杂,大量优质局部最小解散乱且深入地分布于解空间不同区域,增加了算法发现优质解的难度.因此,本文在算法底层设计由多种有效邻域操作(对应多种邻域结构)动态混合并依次执行的启发式搜索,可在算法搜索到达多种邻域结构共同的局部最小解前一直持续向下搜索,从而增强搜索的深度,有利于算法获得优质解.具体而言,本文基于交换、插入、逆序操作(即求解调度问题常用的有效操作)设计两类共计8种邻域操作(LLH<sub>1</sub> ~ LLH<sub>4</sub>对 $\pi$ 进行操作,LLH<sub>5</sub> ~ LLH<sub>8</sub>对 $V_N$ 进行操作),作为低层启发式操作.

1) LLH<sub>1</sub>:加工序列交叉操作,从加工序列中随机选择两位并进行交换.

2) LLH<sub>2</sub>:加工序列前向插入操作,从加工序列中随机选择两位,位置编号大的工件插入到位置编号小的工件之前.

3) LLH<sub>3</sub>:加工序列逆序操作,从加工序列中随机选择两位,将包含所选两位及其之间的工件序列进行逆序排列.

4) LLH<sub>4</sub>:加工序列随机相邻交叉操作,从加工工

序中随机选择一位,向前一位或后一位的工件交换位置,且向前交换与向后交换发生的概率相等.

5) LLH<sub>5</sub>:加工速度对照表交叉操作,随机选择两个工件,将工件在所有机器上的速度对应进行交换.

6) LLH<sub>6</sub>:加工速度对照表前向插入操作,随机选择两个工件,序号大的工件所有加工速度插入到序号小的工件所有加工速度之前.

7) LLH<sub>7</sub>:加工速度对照表逆序操作,随机选择一个工件,将该工件在不同机器上的所有速度进行逆序排列.

8) LLH<sub>8</sub>:加工速度对照表突变操作,随机选择一个工件,将该工件在不同机器上的所有速度进行突变.

**2.4 基于贡献率的高层策略域个体评价方法**

由于高层策略域个体在低层作为独立的启发式算法更新问题域个体,对其评价应结合其搜索性能,在一定程度上可通过问题域个体在操作前后是否改善这一指标衡量策略域个体搜索性能.同时考虑到策略域个体在解空间中的搜索存在随机性,难以使用一次操作前后问题域个体的改善情况准确对策略域个体的搜索性能进行评价.故引入“贡献率”概念,并设计一种基于贡献率的高层策略域个体评价方法.具体描述如下.

step 1: 一个高层策略域个体对应更新  $\theta$  个低层问题域个体,并对更新后所得新解进行评价.

step 2: 分别判断每个问题域个体更新后所得新解与旧解之间的支配关系.  $\omega$  用于计算有多少个问题域个体更新后的新解不被旧解支配.

step 3: 计算  $\psi = \frac{\omega}{\theta}$ , 并将其作为高层策略域个体的评价价值.

**2.5 交叉熵方法**

使用交叉熵方法对高层策略域种群进行优化.令  $Y = (y_1, y_2, \dots, y_q)$  为某个概率空间  $\eta$  的一个随机向量,本文中  $Y$  即高层策略域的一个个体.随机变量  $y_i (i = 1, 2, \dots, 10)$  表示策略域个体的第  $i$  个位按概率  $P_i = (p_{i1}, p_{i2}, \dots, p_{i8})$  在 8 个 LLHs 中选择一个操作,且对于概率  $P_i$  均满足  $\sum_{l=1}^8 p_{il} = 1$ .

第 2.4 节中高层策略域个体的评价函数可等价表示如下:

$$S(Y^*) = \gamma^* = \max_{Y \in \eta} S(Y). \tag{16}$$

其中:  $\eta$  是策略域个体的解空间,  $Y^*$  是最优个体,  $\gamma^*$  是由目标约束  $S$  确定的最优个体所对应的评价价值.令

样本  $Y$  需要服从概率参数为  $\mu$  的概率分布,式(16)中对  $Y^*$  和  $\gamma^*$  的求解可转化为如何确定最优概率分布  $f(Y; \mu)$  的一个概率估计问题:

$$l(\gamma) = P_{\mu}(S(Y) \geq \gamma) = \sum_y I_{\{S(Y) \geq \gamma\}} \cdot f(Y; \mu) = E_{\mu} I_{\{S(Y) \geq \gamma\}}. \tag{17}$$

其中:  $P_{\mu}$  和  $E_{\mu}$  分别是对分布  $f(Y; \mu)$  的概率度量和期望;  $I_{\{S(Y) \geq \gamma\}}$  是对于  $\eta$  上不同  $\gamma$  值 ( $\gamma$  为 2.4 节 step 3 中的  $\psi$ ) 的示性函数集合,  $E_{\mu} I_{\{S(Y) \geq \gamma\}}$  是样本中目标优于  $\gamma$  的期望值.对  $l$  的估计采用重要采样的方法,根据  $\eta$  上可以对  $S(Y)$  接近  $\gamma$  的分布  $g$  进行采样:

$$\hat{l} = \frac{1}{\lambda} \sum_{i=1}^{\lambda} I_{\{S(Y_i) \geq \gamma\}} \frac{f(Y_i; \mu)}{g(Y_i)}. \tag{18}$$

为了估计  $l$ , 此时需要确定概率参数  $\mu$  使分布  $f$  与  $g$  的最优分布  $g^*$  差距最小.衡量两个概率分布的 Kullback-Leible(K-L) 距离以判断他们之间的相似度,最小化 K-L 距离等价于最小化交叉熵<sup>[22]</sup>, 分布  $p$  对  $q$  的交叉熵定义为  $q$  分布的自信息对  $p$  分布的期望:

$$H(p, q) = E_{y \sim p}[-\ln q(y)] = - \int p(y) \ln q(y) dy. \tag{19}$$

这样一来问题最终转化为求概率参数  $\mu$  使来自样本分布  $f(Y; \mu)$  的抽样分布  $g^*$  与  $f(Y; \mu)$  的交叉熵最小化的问题:

$$\mu^* = \arg \min_{\mu} \left[ - \int g^*(y) \ln f(Y; \mu) dy \right]. \tag{20}$$

将式(18)代入(20)可得概率参数的推断:

$$\mu^* = \max_{\mu} E_{\mu} I_{\{S(Y) \geq \gamma\}} \ln f(Y; \mu). \tag{21}$$

由于样本  $Y$  由概率向量  $P_i$  生成,分布  $f(Y; \mu)$  可以有以下形式:

$$f(Y; p) = \prod_{i=1}^{10} \prod_{l=1}^8 p_{il}^{I\{y_i=j\}}. \tag{22}$$

其中:示性函数  $I\{y_i = j\}$  表示策略域个体第  $i$  位是否选择了低层启发式操作  $j$ ,  $p_{il}^{I\{y_i=j\}}$  表示策略域个体第  $i$  位选择低层启发式操作  $j$  的概率.考虑概率  $P_i$  满足  $\sum_{l=1}^8 p_{il} = 1$ , 对式(21)使用拉格朗日乘子法求解

$$\max_p \left[ \min_{\omega_1, \omega_2, \dots, \omega_n} \left[ E_p I_{\{S(Y) \geq \gamma\}} \ln f(Y; \mu) + \sum_{i=1}^{10} \omega_i \left( \sum_{l=1}^8 p_{il} - 1 \right) \right] \right]. \tag{23}$$

对所有  $l = 1, 2, \dots, 8$ , 由式(23)可得

$$\frac{E_p I_{\{S(Y) \geq \gamma\}} I_{\{y_i=j\}}}{p_{il}} + \omega_i = 0. \tag{24}$$

令  $-\omega_i = E_p I_{\{S(Y) \geq \gamma\}}$ , 对于  $\tau$  个有效样本进行离散化处理后,可给出最优  $p_{il}$

$$p_{il} = \frac{E_p I_{\{S(Y) \geq \gamma\}} I_{\{y_i=j\}}}{E_p I_{\{S(Y) \geq \gamma\}}} = \frac{\sum_{v=1}^{\tau} I_{\{S(Y_v) \geq \gamma\}} I_{\{y_{vi}=j\}}}{\sum_{v=1}^{\tau} I_{\{S(Y_v) \geq \gamma\}}}$$

(25)

其中:  $\sum_{v=1}^{\tau} I_{\{S(Y_v) \geq \gamma\}}$  表示  $\tau$  个样本中目标值优于  $\gamma$  的样本总数, 示性函数  $I_{\{y_{vi}=j\}}$  表示当前样本  $v$  中策略域个体第  $i$  位是否选择低层启发式操作  $j$ .

通过式(25)确定的概率参数即可更新高层策略域个体, 具体描述如下:

step 1: 第  $k$  次迭代中策略域个体编码概率分布参数为  $p(k-1)$ , 通过轮盘赌的采样方式随机生成一定数量的个体, 构成本次迭代的一组随机样本  $\{Y_1, Y_2, \dots, Y_n\}$ .

step 2: 由 step 1 中产生的随机样本按式(25)确定本次迭代的概率参数  $p(k)$ . 通过对样本  $\{Y_1, Y_2, \dots, Y_n\}$  的目标函数值  $\{S(Y_1), S(Y_2), \dots, S(Y_n)\}$  ( $\{S(Y_i): 1, 2, \dots, n\}$  对应第2.4节 step 3 中的  $\psi$ ) 排序, 计算样本的  $\rho$  分位数, 使  $P(S(Y)) \leq \gamma_k^* = \rho$ .

step 3: 对于第  $k+1$  次迭代策略域个体编码的概率参数, 引入平滑系数  $\alpha$  的更新策略, 如下所示:

$$p(k) = \alpha p(k) + (1 - \alpha) p(k - 1)$$

(26)

### 2.6 基于非关键路径的节能策略

对于 GDPFSP\_FPT 中的每一个工厂而言, 从第 1 台加工机器开始加工到最后一台机器完工的最长路径, 称为关键路径. 关键路径决定了当前工厂的模糊最大完工时间(即  $\tilde{C}_f$ ), 第 2.1 节中所举示例的关键路径如图 5 所示. 通过保持关键路径上操作的加工速度不变, 对非关键路径上操作的加工速度进行调节, 可实现在模糊最大完工时间不变的前提下尽可能降低模糊总能耗的目的.

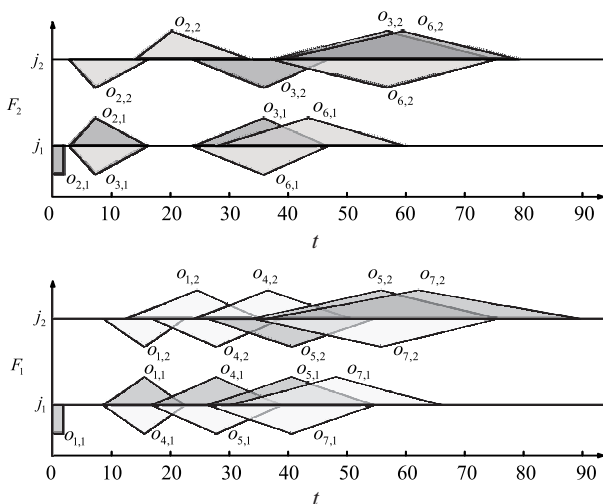


图 5 关键路径

节能策略具体描述如下:

step 1: 判断该操作的加工速度是否等于最低档位速度. 若等于则不对速度进行调节, 输出当前加工速度, 并终止循环; 否则进行下一步.

step 2: 将当前该操作的加工速度降低一个档位, 并判断是否影响关键路径. 若影响则输出本次调节前的加工速度, 并终止循环; 否则转到 step 1.

### 2.7 算法流程

根据上述算法描述, 整个算法具体流程描述如下.

step 1: 初始化高层策略域与低层问题域种群.

step 2: 对问题域种群进行评价.

step 3: 对问题域个体的支配关系进行判断, 并计算个体对应的支配等级与拥挤距离. 对问题域种群排序后, 选择前  $\text{popsize} \times 15\%$  个个体构成优秀个体集合, 并对优秀个体集合中个体进行调速操作.

step 4: 使用策略域个体每一位对应的低层启发式操作依次更新对应问题域个体. 每执行完一个低层启发式操作, 则对操作前后的两个解进行比较. 若新解支配旧解, 则使用新解代替旧解继续执行剩余低层启发式操作; 否则, 有一定概率使用新解代替旧解, 并继续执行剩余低层启发式操作, 实现对问题域种群进行更新.

step 5: 将更新后的问题域种群与优秀个体集合合并, 排序后选择前  $\text{popsize}$  个个体组成新问题域种群.

step 6: 对策略域种群进行评价, 并使用交叉熵方法对策略域种群进行更新.

step 7: 使用由问题域种群得到的非支配解集更新 Pareto 档案集.

step 8: 判断是否满足终止条件, 若不满足则转到 step 2, 否则终止循环.

## 3 实验设计与分析

由于目前没有适合 GDPFSP\_FPT 的标准算例, 本文所有测试问题的加工时间均在由文献[2]为解决 DPFSP 所提供测试算例的基础上生成.

### 3.1 参数设置

机器空载时的单位能耗  $\varepsilon$  设置为 1. HHCE 中问题域种群容量  $\text{popsize}$  设置为 100, 问题域中精英个体所占比例  $\theta_\alpha$  设置为 0.2, 低层启发式操作的接受概率  $\varphi$  设置为 0.3, 统计样本的  $\rho$  分位数与策略域种群规模的比值  $\theta_\rho$  设置为 0.8, 每个策略域个体更新问题域个体的数量  $\vartheta$  设置为 5, 平滑系数  $\alpha$  设置为 0.3.

3.2 仿真结果与比较

对于不同问题, 设定每种算法的运行时间均为  $1.5 \times N$  s. 所有测试问题均进行20次独立实验. 每个问题对应最优结果用粗体表示.

为评价各算法性能, 采用2个评价指标对不同算法所得非支配解集  $S_j$  进行评价. 第1个指标如下:

$$R\_NDS(S_j) = \frac{|S_j - \{x \in S_j | \exists y \in S : y \prec x\}|}{|S_j|}, \tag{27}$$

其衡量的是算法所得  $S_j$  的整体质量; 第2个指标为

$$NDS\_NUM(S_j) = |S_j - \{x \in S_j | \exists y \in S : y \prec x\}|. \tag{28}$$

其衡量的是算法获取非支配解的能力. 显然, 这两个指标的值越大,  $S_j$  表现越好.

在式(27)和(28)中:  $S$  表示所有算法产生的非支配解集的集合,  $y \prec x$  表示解  $x$  被  $y$  支配,  $|S_j|$  表示非支配解集中解的数量.

为验证 HHCE 中选择 CE 方法作为高层策略的有效性, 将 HHCE 与采用遗传算法 (genetic algorithm, GA) 作为高层策略的超启发式遗传算法 (hyperheuristic genetic algorithm, HHGA) 进行比较. 这两个算法除高层策略不同之外, 其他部分均相同. 比较结果如表2所示.

由表2可知, HHCE 在绝大部分问题上的测试结果都明显优于 HHGA, 这验证了 CE 方法作为超启发式算法中高层启发式策略的有效性. 实际上, CE 方法作为基于概率模型的方法, 可一定程度上避免 GA 这类传统进化方法普遍存在的较优解模式破坏的问题<sup>[23]</sup>, 从而能更好地保留较优解信息并合理引导搜索方向.

为验证第2.6节所提节能策略的有效性, 将 HHCE 与不带节能策略的 HHCE (HHCE\_noES) 进行比较, 比较结果如表3所示.

表2 算法比较(HHGA与HCE)

$N \times M \times F$	HHGA		HHCE	
	$R\_N$	$N\_N$	$R\_N$	$N\_N$
$20 \times 5 \times 2$	5.8	0.389	<b>27.2</b>	<b>0.972</b>
$20 \times 5 \times 3$	9.2	0.369	<b>33.0</b>	<b>0.969</b>
$20 \times 5 \times 4$	2.4	0.115	<b>34.0</b>	<b>0.945</b>
$20 \times 10 \times 2$	5.0	0.243	<b>26.2</b>	<b>0.874</b>
$20 \times 10 \times 3$	4.2	0.361	<b>22.8</b>	<b>0.915</b>
$20 \times 10 \times 4$	4.6	0.248	<b>28.0</b>	<b>0.880</b>
$20 \times 20 \times 2$	6.4	0.246	<b>29.6</b>	<b>0.819</b>
$20 \times 20 \times 3$	4.0	0.131	<b>33.4</b>	<b>0.891</b>
$20 \times 20 \times 4$	2.8	0.181	<b>28.2</b>	<b>0.936</b>
$50 \times 5 \times 2$	4.0	0.627	<b>20.2</b>	<b>0.873</b>
$50 \times 5 \times 3$	3.2	0.276	<b>23.0</b>	<b>0.917</b>
$50 \times 5 \times 4$	4.4	0.283	<b>29.0</b>	<b>0.948</b>
$50 \times 10 \times 2$	4.8	0.143	<b>31.2</b>	<b>0.930</b>
$50 \times 10 \times 3$	2.6	0.157	<b>25.8</b>	<b>0.907</b>
$50 \times 10 \times 4$	3.2	0.132	<b>33.2</b>	<b>0.944</b>
$50 \times 20 \times 2$	3.4	0.165	<b>33.0</b>	<b>0.911</b>
$50 \times 20 \times 3$	2.4	0.094	<b>30.0</b>	<b>0.882</b>
$50 \times 20 \times 4$	3.6	0.103	<b>31.6</b>	<b>0.941</b>
$100 \times 5 \times 2$	2.2	0.127	<b>34.0</b>	<b>0.974</b>
$100 \times 5 \times 3$	4.0	0.147	<b>38.0</b>	<b>0.940</b>
$100 \times 5 \times 4$	2.8	0.123	<b>35.2</b>	<b>0.948</b>
$100 \times 10 \times 2$	5.0	0.168	<b>24.5</b>	<b>0.863</b>
$100 \times 10 \times 3$	2.0	0.061	<b>32.8</b>	<b>0.940</b>
$100 \times 10 \times 4$	3.0	0.109	<b>28.0</b>	<b>0.896</b>
$100 \times 20 \times 2$	1.6	0.064	<b>29.0</b>	<b>0.975</b>
$100 \times 20 \times 3$	4.2	0.154	<b>30.2</b>	<b>0.896</b>
$100 \times 20 \times 4$	6.6	0.238	<b>19.6</b>	<b>0.752</b>

表3 算法比较(HHCE\_noES与HHCE)

$N \times M \times F$	HHCE_noES		HHCE	
	$R\_N$	$N\_N$	$R\_N$	$N\_N$
$20 \times 5 \times 2$	22.6	0.679	<b>30.6</b>	<b>0.931</b>
$20 \times 5 \times 3$	26.0	0.863	<b>29.2</b>	<b>0.889</b>
$20 \times 5 \times 4$	25.6	0.664	<b>28.2</b>	<b>0.889</b>
$20 \times 10 \times 2$	22.6	0.767	<b>28.2</b>	<b>0.951</b>
$20 \times 10 \times 3$	23.6	0.692	<b>26.0</b>	<b>0.901</b>
$20 \times 10 \times 4$	22.8	0.704	<b>28.0</b>	<b>0.902</b>
$20 \times 20 \times 2$	22.4	0.747	<b>34.8</b>	<b>0.943</b>
$20 \times 20 \times 3$	22.6	0.720	<b>33.0</b>	<b>0.923</b>
$20 \times 20 \times 4$	13.6	0.612	<b>33.4</b>	<b>0.908</b>
$50 \times 5 \times 2$	21.0	0.722	<b>30.4</b>	<b>0.975</b>
$50 \times 5 \times 3$	23.6	0.832	<b>24.6</b>	<b>0.953</b>
$50 \times 5 \times 4$	22.4	<b>0.918</b>	<b>23.0</b>	0.892
$50 \times 10 \times 2$	17.8	0.714	<b>28.4</b>	<b>0.925</b>
$50 \times 10 \times 3$	14.2	0.643	<b>28.2</b>	<b>0.926</b>
$50 \times 10 \times 4$	13.0	0.595	<b>26.2</b>	<b>0.858</b>
$50 \times 20 \times 2$	14.8	0.624	<b>29.0</b>	<b>0.886</b>
$50 \times 20 \times 3$	12.8	0.568	<b>30.4</b>	<b>0.958</b>
$50 \times 20 \times 4$	10.0	0.424	<b>28.8</b>	<b>0.922</b>
$100 \times 5 \times 2$	20.2	0.749	<b>33.8</b>	<b>0.921</b>
$100 \times 5 \times 3$	16.2	0.624	<b>34.8</b>	<b>0.937</b>
$100 \times 5 \times 4$	17.6	0.610	<b>31.8</b>	<b>0.934</b>
$100 \times 10 \times 2$	14.4	0.560	<b>35.4</b>	<b>0.908</b>
$100 \times 10 \times 3$	13.4	0.516	<b>34.2</b>	<b>0.949</b>
$100 \times 10 \times 4$	13.4	0.569	<b>31.0</b>	<b>0.926</b>
$100 \times 20 \times 2$	15.2	0.557	<b>30.2</b>	<b>0.898</b>
$100 \times 20 \times 3$	16.2	0.617	<b>31.0</b>	<b>0.945</b>
$100 \times 20 \times 4$	13.4	0.662	<b>28.4</b>	<b>0.942</b>

由表3可知,HHCE在绝大部分问题上的测试结果都明显优于HHCE\_noES,验证了本文所提基于非关键路径的节能策略的有效性。

为验证HHCE的有效性,将HHCE与SPEA2<sup>[24]</sup>、MOBSO<sup>[13]</sup>进行比较。SPEA2是一种广泛应用于求解多种车间调度问题的经典多目标算法。MOBSO是近年来求解分布式流水线绿色调度问题的有效算法。各算法比较结果如表4所示。

表4 算法比较结果(SPEA2、MOBSO与HHCE)

$N \times M \times F$	SPEA2		MOBSO		HHCE	
	$R_N$	$N_N$	$R_N$	$N_N$	$R_N$	$N_N$
20 × 5 × 2	0.4	0.050	20.8	0.497	<b>30.2</b>	<b>0.933</b>
20 × 5 × 3	0.0	0.000	25.2	0.614	<b>31.6</b>	<b>0.969</b>
20 × 5 × 4	0.0	0.000	28.6	0.640	<b>32.6</b>	<b>0.982</b>
20 × 10 × 2	0.2	0.033	15.6	0.458	<b>22.2</b>	<b>0.850</b>
20 × 10 × 3	0.0	0.000	<b>26.8</b>	0.656	24.0	<b>0.910</b>
20 × 10 × 4	0.2	0.050	21.4	0.562	<b>33.2</b>	<b>0.889</b>
20 × 20 × 2	0.2	0.067	22.8	0.549	<b>37.0</b>	<b>0.906</b>
20 × 20 × 3	0.0	0.000	18.2	0.555	<b>41.0</b>	<b>0.876</b>
20 × 20 × 4	0.4	0.067	15.8	0.452	<b>33.2</b>	<b>0.893</b>
50 × 5 × 2	0.0	0.000	18.0	0.782	<b>27.2</b>	<b>0.971</b>
50 × 5 × 3	0.0	0.000	25.8	0.928	<b>26.4</b>	<b>0.969</b>
50 × 5 × 4	0.2	0.029	17.2	0.803	<b>23.2</b>	<b>0.953</b>
50 × 10 × 2	0.8	0.207	14.2	0.746	<b>29.0</b>	<b>0.924</b>
50 × 10 × 3	0.6	0.147	11.8	0.692	<b>29.8</b>	<b>0.921</b>
50 × 10 × 4	0.4	0.100	13.6	0.602	<b>31.6</b>	<b>0.905</b>
50 × 20 × 2	0.6	0.173	7.2	0.610	<b>30.8</b>	<b>0.889</b>
50 × 20 × 3	0.4	0.057	9.2	0.557	<b>22.8</b>	<b>0.894</b>
50 × 20 × 4	0.6	0.167	9.8	0.544	<b>29.8</b>	<b>0.914</b>
100 × 5 × 2	0.0	0.000	19.6	<b>0.949</b>	<b>32.4</b>	0.916
100 × 5 × 3	1.0	0.210	9.4	0.657	<b>31.2</b>	<b>0.894</b>
100 × 5 × 4	0.6	0.113	12.8	0.853	<b>27.8</b>	<b>0.903</b>
100 × 10 × 2	0.6	0.162	7.4	0.760	<b>25.2</b>	<b>0.892</b>
100 × 10 × 3	0.6	0.120	6.4	0.543	<b>30.0</b>	<b>0.897</b>
100 × 10 × 4	0.6	0.117	9.6	0.564	<b>21.8</b>	<b>0.935</b>
100 × 20 × 2	0.0	0.000	10.8	<b>0.985</b>	<b>28.0</b>	0.922
100 × 20 × 3	0.2	0.050	8.0	0.563	<b>33.0</b>	<b>0.946</b>
100 × 20 × 4	0.4	0.090	6.4	0.532	<b>31.4</b>	<b>0.935</b>

由表4可知,HHCE在绝大部分问题上的测试结果都明显优于比较算法,表明HHCE是求解GDPFSP\_FPT的有效算法。

SPEA2和MOBSO这些常规群智能算法每代均是采用数个固定的交叉、变异等邻域操作生成新种群以实现搜索,这使其实际搜索深度较为有限,难以到达复杂解空间中真正存在优质解的区域。HHCE利

用高层的概率模型动态控制低层每个个体执行包含多个邻域操作的启发式搜索,这有利于引导算法在多个不同区域进行较深入搜索,从而较易发现复杂解空间中的优质解。因此,HHCE可在上述测试问题中取得较好结果。

### 4 结论

本文在传统分布式流水线调度问题(distributed permutation flow-shop scheduling problem,DPFSP)的基础上,进一步考虑实际生产中普遍存在的加工时间不确定性,首次建立模糊分布式流水线绿色调度问题(green distributed permutation flow-shop scheduling problem with fuzzy processing time,GDPFSP\_FPT)的模型。该模型采用模糊数表示加工时间,并以同时最小化模糊最大完工时间和模糊总能耗为优化目标。针对GDPFSP\_FPT,本文提出了一种超启发式交叉熵算法(hyper-heuristic cross-entropy algorithm,HHCE)进行求解。HHCE具有如下优点:1)选用一种更为合理的模糊数排序准则,以协助算法在搜索过程中较准确发现优质解区域;2)在算法高层利用交叉熵(cross-entropy,CE)方法动态确定低层个体每代具体执行的启发式搜索,可推动算法到达较多的优质解区域;3)采用基于非关键路径的节能策略,能增强算法获取低能耗非劣个体或解的能力。仿真实验和算法比较验证了HHCE是求解GDPFSP\_FPT的有效算法。未来研究将把HHCE扩展用于求解不确定分布式装配线调度问题。

### 参考文献(References)

- [1] 顾幸生. 不确定性条件下的生产调度[J]. 华东理工大学学报, 2000, 26(5): 441-446.  
(Gu X S. A survey of production scheduling under uncertainty[J]. Journal of East China University of Science and Technology, 2000, 26(5): 441-446.)
- [2] Naderi B, Ruiz R. The distributed permutation flowshop scheduling problem[J]. Computers & Operations Research, 2010, 37(4): 754-768.
- [3] Fernandez-Viagas V, Framinan J M. A bounded-search iterated greedy algorithm for the distributed permutation flowshop scheduling problem[J]. International Journal of Production Research, 2015, 53(4): 1111-1123.
- [4] Bargaoui H, Belkahla Driss O, Ghédira K. A novel chemical reaction optimization for the distributed permutation flowshop scheduling problem with makespan criterion[J]. Computers & Industrial Engineering, 2017, 111(1): 239-250.
- [5] Zhang G A, Xing K Y, Cao F. Discrete differential

- evolution algorithm for distributed blocking flowshop scheduling with makespan criterion[J]. *Engineering Applications of Artificial Intelligence*, 2018, 76: 96-107.
- [6] Chen J F, Wang L, He X, et al. A probability model-based memetic algorithm for distributed heterogeneous flow-shop scheduling[C]. *2019 IEEE Congress on Evolutionary Computation (CEC)*. New York: IEEE, 2019: 411-418.
- [7] Palacios J J, González-Rodríguez I, Vela C R, et al. Swarm lexicographic goal programming for fuzzy open shop scheduling[J]. *Journal of Intelligent Manufacturing*, 2015, 26(6): 1201-1215.
- [8] 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.
- [9] Liu G S, Zhou Y, Yang H D. Minimizing energy consumption and tardiness penalty for fuzzy flow shop scheduling with state-dependent setup time[J]. *Journal of Cleaner Production*, 2017, 147: 470-484.
- [10] Sun L, Lin L, Gen M S, et al. A hybrid cooperative co-evolution algorithm for fuzzy flexible job shop scheduling[J]. *IEEE Transactions on Fuzzy Systems*, 2019, 27(5): 1008-1022.
- [11] 王凌, 王晶晶, 吴楚格. 绿色车间调度优化研究进展[J]. *控制与决策*, 2018, 33(3): 385-391.  
(Wang L, Wang J J, Wu C G. *Advances in green shop scheduling and optimization*[J]. *Control and Decision*, 2018, 33(3): 385-391.)
- [12] Deng J, Wang L, Wu C G, et al. A competitive memetic algorithm for carbon-efficient scheduling of distributed flow-shop[C]. *International Conference on Intelligent Computing*. Cham: Springer, 2016: 476-488.
- [13] Fu Y P, Tian G D, Fathollahi-Fard A M, et al. Stochastic multi-objective modelling and optimization of an energy-conscious distributed permutation flow shop scheduling problem with the total tardiness constraint[J]. *Journal of Cleaner Production*, 2019, 226: 515-525.
- [14] Cowling P, Kendall G, Soubeiga E. A hyperheuristic approach to scheduling a sales summit[C]. *International Conference on the Practice and Theory of Automated Timetabling*. Berlin: Springer, 2000: 176-190.
- [15] Wang H. An adaptive hyper-heuristics genetic algorithm for stochastic job shop scheduling problem[C]. *Applied Mechanics and Materials*. Switzerland: Trans Tech Publications, Ltd, 2014: 3956-3960.
- [16] Lin J, Wang Z J, Li X D. A backtracking search hyper-heuristic for the distributed assembly flow-shop scheduling problem[J]. *Swarm and Evolutionary Computation*, 2017, 36: 124-135.
- [17] Rubinstein R Y. Optimization of computer simulation models with rare events[J]. *European Journal of Operational Research*, 1997, 99(1): 89-112.
- [18] Santosa B, Budiman M A, Wiratno S E. A cross entropy-genetic algorithm for m-machines no-wait job-shop scheduling problem[J]. *Journal of Intelligent Learning Systems and Applications*, 2011, 3(3): 171-180.
- [19] 王桂荣, 李歧强, 丁然, 等. 加工时间不确定的炼钢连铸生产调度串级交叉熵算法[J]. *控制与决策*, 2016, 31(7): 1153-1160.  
(Wang G R, Li Q Q, Ding R, et al. *Cascade cross entropy algorithm for steelmaking-continuous casting production scheduling with uncertain processing time*[J]. *Control and Decision*, 2016, 31(7): 1153-1160.)
- [20] 李尚函, 胡蓉, 钱斌, 等. 超启发式遗传算法求解模糊柔性作业车间调度[J]. *控制理论与应用*, 2020, 37(2): 316-330.  
(Li S H, Hu R, Qian B, et al. *Hyper-heuristic genetic algorithm for solving fuzzy flexible job shop scheduling problem*[J]. *Control Theory & Application*, 2020, 37(2): 316-330.)
- [21] Sakawa M, Kubota R. Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms[J]. *European Journal of Operational Research*, 2000, 120(2): 393-407.
- [22] Boer P T D, Kroese D P, Mannor S, et al. A tutorial on the cross-entropy method[J]. *Annals of Operations Research*, 2005, 134(1): 19-67.
- [23] 杨海军, 李建武, 李敏强. 进化算法的模式、涌现与困难性研究[M]. 北京: 科学出版社, 2012: 89-102.  
(Yang H J, Li J W, Li M Q. *Evolutionary algorithms: Schema, emergence and hardness*[M]. Beijing: Science Press, 2012: 89-102.)
- [24] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm[J]. *TIK-report*, 2001: 103.

### 作者简介

钱斌(1976—), 男, 教授, 博士生导师, 从事智能优化调度理论与方法等研究, E-mail: bin.qian@vip.163.com;

余明哲(1994—), 男, 硕士生, 从事优化调度与智能算法的研究, E-mail: shemingzhe@163.com;

胡蓉(1974—), 女, 副教授, 博士生, 从事优化方法与决策支持系统的研究, E-mail: ronghu@vip.163.com;

郭宁(1983—), 女, 实验师, 博士生, 从事物流优化调度与智能算法的研究, E-mail: 124328466@qq.com;

向凤红(1964—), 男, 教授, 博士, 从事智能优化方法的研究, E-mail: xiangfh5447@sina.com.

(责任编辑: 闫妍)