第 37 卷第 10 期
2020 年 10 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 37 No. 10
Oct. 2020

混合交叉熵算法求解模糊分布式装配流水线低碳调度问题

佘明哲[1,2], 钱　斌[1,2†], 胡　蓉[1,2], 吴丽萍[1], 向凤红[1]

(1. 昆明理工大学 信息工程与自动化学院, 云南 昆明 650500;

2. 昆明理工大学 云南省人工智能重点实验室, 云南 昆明 650500)

摘要: 本文针对实际生产过程中普遍存在的不确定性, 采用模糊数表示工件的加工时间与产品的装配时间, 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, 建立模糊分布式装配流水线低碳调度问题(FDAPFLSP)的模型, 进而提出一种混合交叉熵算法(HCEA)进行求解. 首先, 通过分析现有三角模糊数排序准则特点, 并考虑生产调度问题的基本约束, 设计一种实用的三角模糊数排序修正准则. 其次, 为增强算法性能, 设计一种自适应变邻域局部搜索以实现对解空间不同区域的有效搜索. 最后, 仿真实验与算法对比验证HCEA可有效求解FDAPFLSP.

关键词: 分布式装配流水线调度; 模糊加工时间; 模糊装配时间; 低碳; 多目标优化; 交叉熵算法

引用格式: 佘明哲, 钱斌, 胡蓉, 等. 混合交叉熵算法求解模糊分布式装配流水线低碳调度问题. 控制理论与应用, 2020, 37(10): 2081 – 2092

DOI: 10.7641/CTA.2020.00038

Hybrid cross-entropy algorithm for fuzzy distributed assembly permutation flow-shop low-carbon scheduling problem

SHE Ming-zhe[1,2], QIAN Bin[1,2†], HU Rong[1,2], WU Li-ping[1], XIANG Feng-hong[1]

(1. Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming Yunnan 650500, China;

2. Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming Yunnan 650500, China)

Abstract: Considering the widely existing uncertainty in the real-world production process, this paper uses fuzzy numbers to represent each job's processing time and each product's assembly time, and constructs a model for the fuzzy distributed assembly permutation flow-shop low-carbon scheduling problem (FDAPFLSP), whose criteria are the minimization of both the fuzzy maximum completion time and the fuzzy total energy consumption. Then, a hybrid cross-entropy algorithm (HCEA) is proposed for solving the FDAPFLSP. Firstly, a practical ranking correction rule of triangular fuzzy number is designed via analyzing the characteristics of the commonly used ranking rules of triangular fuzzy number and considering the basic constraints of the production scheduling problem. Secondly, HCEA adopts variable neighborhood local search with adaptive selection probability, which can efficiently search different regions in solution space and further enhance the performance of the algorithm. Finally, simulations and comparisons demonstrate that HCEA can effectively solve FDAPFLSP.

Key words: distributed assembly permutation flow-shop scheduling; fuzzy processing time; fuzzy assembly time; low-carbon; multi-objective optimization; cross-entropy algorithm

Citation: SHE Mingzhe, QIAN Bin, HU Rong, et al. Hybrid cross-entropy algorithm for fuzzy distributed assembly permutation flow-shop low-carbon scheduling problem. *Control Theory & Applications*, 2020, 37(10): 2081 – 2092

1 引言

随着市场竞争的加剧以及经济全球化趋势的深入发展, 具有多个生产中心的制造方式变得越来越普遍, 企业可从不同的供货商处采购产品所需部件, 后对其组装. 这种方式能够在有效地把控产品质量的同时降低生产成本与管理风险. 因此, 对分布式装配系统展开研究具有重要意义. 同时, 由于实际生产过程大多存在不确定性, 这类因素的存在可能导致在确定条件

收稿日期: 2020−01−15; 录用日期: 2020−07−06.

†通信作者. E-mail: bin.qian@vip.163.com; Tel.: +86 13312529481.

本文责任编委: 王凌.

国家自然科学基金项目(51665025, 61963022)资助.

Supported by the National Natural Science Foundation of China (51665025, 61963022).

假设下得到的优质调度方案在实际生产过程中变得不可行. 所以利用模糊理论中的模糊数表征不确定的加工和装配时间, 有助于更为客观地描述实际生产过程, 并为生产过程提供指导. 另外, 针对全球日益严峻的环境问题, 绿色制造开始逐渐引起企业的重视[1–2]. 因此, 研究模糊分布式装配流水线低碳调度问题(fuzzy distributed assembly permutation flow-shop low-carbon scheduling problem, FDAPFLSP)具有现实的经济价值和深远的社会意义. 在计算复杂度上, 分布式装配流水线调度问题(distributed assembly permutation flow-shop scheduling problem, DAPFSP)已被证明具有非确定多项式难(non-deterministic polynomial hard, NP-hard)的属性[3], 而该问题又约归为FDAPFLSP, 故FDAPFLSP属于NP-hard问题. 因此, 研究FDAPFLSP问题及其求解算法具有重要的理论价值和实际意义.

在计算模糊生产调度问题的不同工件排列或解对应的目标函数值时, 需要对模糊加工和装配时间进行加、减、乘、取大等运算; 同时评价不同解的优劣时, 需要对不同解的目标函数值进行比较运算. 其中加、减、乘运算可通过模糊数学的相关定义直接进行, 取大、比较运算可通过对模糊数进行排序实现. 现已有学者对模糊生产调度问题中模糊数排序准则进行研究. Sakawa等[4]设计一种基于隶属度参数值的三角模糊数排序准则. Sakawa等[5]在文献[4]的基础上, 设计3条排序准则用以判断模糊数的大小关系. Lei[6]证明文献[5]所提排序准则优于文献[4]所提准则. Li 等[7]在充分考虑取大操作的近似误差和模糊度基础上, 设计一种新颖的三角模糊数排序准则. 由文献调研可知, 模糊生产调度问题的已有研究多采用文献[5]所提准则对三角模糊数的大小关系进行判断. 且上述3种排序准则均从排序的准确性角度对准则进行改进, 并未考虑生产调度问题特性, 故直接使用上述文献所提准则, 易出现违反生产调度问题约束的情况, 进而导致算法在违反问题约束的解空间区域周围进行搜索, 影响求解过程中的搜索方向与最终所得解的质量, 甚至使所得调度优质解在实际生产过程中变得不可行. 因此, 需要在原有模糊数排序准则的基础上进行修正, 以避免此类情况的出现. 目前, 尚无针对此不足对模糊数排序准则进行改进的研究.

在分布式装配流水线调度问题研究方面, Hatami等[3]首次提出了该问题, 以最小化最大完工时间为优化目标建立了其混合整数线性规划(mixed integer linear program, MILP)模型, 同时提出多个启发式方法用于构造初始解, 并设计变邻域下降算法(variable neighborhood descent algorithm, VND)进行求解. Lin等[8]针对优化目标为最小化最大完工时间的分布式装配流水线调度问题, 设计一种回溯搜索超启发式(backtracking search hyper-heuristic, BSHH)算法进行求解, 该算法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. Sang等[9]针对优化目标为总流经时间的分布式装配流水线调度问题, 设计3种离散杂草入侵优化(discrete invasive weed optimization, DIWO)算法进行求解. Ferone等[10]针对优化目标为最小化最大完工时间的分布式装配流水线调度问题, 设计一种有偏迭代局部搜索(biased-based iterated local search, BR–ILS)算法进行求解, 该算法采用引入有偏随机行为的启发式算法构造初始解以提升其质量. 显然, 分布式装配流水线调度问题正成为当前的研究热点.

生产过程存在多种不确定性因素[11], 譬如工艺不稳定、设备老化、操作人员不同等导致工件或工序加工时间难以提前确定. 因此, 采用模糊数表示加工和装配时间, 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. Yang等[12]针对优化目标为最小化模糊最大完工时间和最大化平均满意度的带堵塞模糊流水车间调度问题, 设计了一种混合多目标灰狼优化(hybrid multi-objective gray wolf optimization, HMOGWO)算法进行求解, 该算法在构造初始解阶段设计了一种以最大化平均满意度为导向的启发式算法, 与PFE启发式算法共同使用以提升初始解的质量; 另外, 在局部搜索阶段基于insert邻域结构设计了两种局部搜索操作以提升算法的搜索能力. Li等[13]针对优化目标为最小化模糊最大完工时间的带资源消耗约束的模糊同类机调度问题, 设计了一种简化群优化(simplified swarm optimization, SSO)算法进行求解, 该算法引入了惩罚函数以引导算法向满足资源消耗约束的解空间区域的方向进行搜索. Zhong等[14]针对优化目标为同时最小化模糊最大完工时间和模糊最大机器负载、最大化加权满意度的模糊柔性作业车间调度问题, 设计了一种改进人工蜂群(modified artificial bee colony, MABC)算法进行求解, 该算法设计了一种基于变邻域搜索的局部搜索操作, 以实现对不同解空间区域搜索的目的. 然而, 尚无人对模糊分布式装配流水线调度问题进行研究.

与传统调度问题相比, 低碳调度问题通过资源分配、操作排序和运行模式的综合优化, 实现节能减排、降本增效, 求解难度更大, 也更具工程意义和理论价值[15]. 在分布式车间低碳调度问题研究方面, Fu等[16]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低碳调度问题, 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, 该算法通过判断种群中个体的支配等级对其进行分组, 然后采用二值选择法确定一或两组, 对应进行

不同操作以生成新个体. Chen 等[17]针对优化目标为同时最小化最大完工时间和总能耗的分布式零空闲流水线低碳调度问题, 设计了一种协同进化算法 (collaborative optimization algorithm, COA) 进行求解, 该算法采用两种启发式算法构造初始解以提升其质量, 在局部搜索阶段多种操作自适应更新选择概率以增强算法搜索能力. 目前没有分布式装配流水线低碳调度问题 (distributed assembly permutation flow-shop low-carbon scheduling problem, DAPFLSP)方面的研究报道. 由上述文献调研可知, 尚无求FDAPFLSP这类重要问题的文献报道, 故对其开展建模和算法研究具有较大理论和实际意义.

交叉熵算法 (cross-entropy algorithm, CEA) 是由Rubinstein[18]提出的一种小概率事件估计算法. 该算法利用稀有事件(即较优解)信息更新相应的概率模型参数, 并使用有效采样方法对概率模型采样以生成新种群, 进而引导搜索方向. CEA采用正反馈机制使得较优解的模式信息得以更多积累, 有利于引导算法到达问题解空间中存在优质解的区域. 这使得算法具有较强的全局搜索能力, 从而在多类优化问题上得到成功应用. 譬如, Caserta等[19]针对优化目标为最大化装包物品数量的带装包费用的整数背包问题, 设计了混合交叉熵算法(hybrid cross entropy algorithm, HCEA)进行求解. 该算法采用CEA确定放入背包的物品种类, 进而利用动态规划算法确定各类物品的数量. Jędrzejowicz等[20]针对优化目标为最小化最大完工时间的同构并行机调度问题, 设计了基于交叉熵的群学习算法 (cross-entropy-based population-learning algorithm, CPLA)进行求解. 该算法利用CEA和禁忌搜索算法共同产生种群, 进而采用岛屿进化算法进一步提高种群质量. Santosa等[21]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, 设计了一种混合交叉熵遗传算法 (hybrid cross-entropy genetic algorithm, CEGA) 进行求解. 该算法在CEA基础上, 进一步利用遗传算法的交叉、变异操作来实现对解空间中更多不同区域的搜索. Wang等[22]针对优化目标为最小化总能耗的炼钢–连铸生产调度问题, 设计了改进交叉熵算法 (improved cross entropy algorithm, ICEA)进行求解. 该算法在对CEA概率矩阵采样时加入启发式规则以改进生成解的质量, 并对部分精英个体执行基于交换操作的局部搜索. Li等[23]针对高维函数优化问题设计了一种混合交叉熵萤火虫算法 (cross-entropy firefly algorithm, CEFA). 该算法采用CEA和萤火虫算法两套不同搜索机制进行协同搜索, 以提高算法对解空间不同区域的搜索能力. 从CEA在优化问题上的研究现状来看, 将CEA与其他有效的算法和操作合理混合, 可丰富搜索行为而提升算法性能, 是设计有效CEA的关键. 目前尚无CEA求解分布式生产调度问题的相关研究.

本文研究FDAPFLSP的建模与求解. 首先, 在模糊数排序准则方面, 分析目前常用排序准则的特点, 充分考虑生产调度问题约束, 在原有排序准则的基础上设计一种实用的三角模糊数排序修正准则. 其次, 在问题建模方面, 针对实际生产中广泛存在的加工和装配时间不确定性, 采用三角模糊数进行表征, 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FDAPFLSP问题模型. 然后, 在算法设计方面, 考虑到CEA具有较强全局搜索能力但收敛较慢, 故引入局部搜索并提出一种混合交叉熵算法 (hybrid cross-entropy algorithm, HCEA)进行求解. 在HCEA中, 全局搜索阶段采用CEA的概率模型学习和积累部分较优排列的信息, 进而通过采样该模型以实现对解空间中优质区域的搜索; 局部搜索阶段提出基于多种邻域操作的自适应变邻域局部搜索, 在算法进化过程中根据各邻域操作的贡献率自适应选择合适的邻域操作进行搜索, 可对全局搜索发现的优质区域进行更为细致的搜索, 从而增强算法性能. 最后, 通过仿真实验和算法比较验证所提算法的有效性.

2　模糊分布式装配流水线低碳调度问题

2.1　符号定义

本文所涉及的数学符号及定义如表1所示.

表 1　符号表

Table 1　Notations

符号	说　明
f	工厂序列 $f=\{1,2,\cdots,F\}$
j	机器序列 $j=\{1,2,\cdots,M\}$
i	工件序列 $i=\{1,2,\cdots,N\}$
h	产品序列 $h=\{1,2,\cdots,Q\}$
popsize	种群容量
$o_{i,j}$	工件i在机器j上的操作
$\tilde{p}_{i,j}$	操作$o_{i,j}$的加工时间
N_h	装配产品P_h所需工件数
$\tilde{p}_h^{\mathrm{A}}$	装配产品P_h所需时间
π_h^{A}	装配产品P_h的工件加工序列$\pi_h^{\mathrm{A}}=\{\pi_1^{\mathrm{A}},\pi_2^{\mathrm{A}},\cdots,\pi_{N_h}^{\mathrm{A}}\}$
π	工件加工序列$\pi=\{\pi_1,\pi_2,\cdots,\pi_N\}$
π^{f}	工厂f的工件加工序列$\pi^{\mathrm{f}}=\{\pi_1^{\mathrm{f}},\pi_2^{\mathrm{f}},\cdots,\pi_{N_{\mathrm{f}}}^{\mathrm{f}}\}$
π^{A}	装配机器上的产品序列$\pi^{\mathrm{A}}=\{\pi_1^{\mathrm{A}},\pi_2^{\mathrm{A}},\cdots,\pi_Q^{\mathrm{A}}\}$
$\tilde{t}_{\mathrm{off}}$	加工机器关机所需时间
$\tilde{t}_{\mathrm{on}}$	加工机器开机所需时间
T	加工机器可执行开关机次数
$\tilde{S}_{i,j}$	操作$o_{i,j}$的开工时间
$\tilde{C}_{i,j}$	操作$o_{i,j}$的完工时间
$\tilde{C}_h$	产品P_h所需全部工件加工完成后的最大完工时间
$\varepsilon_{\mathrm{idle}}$	加工机器单位时间的空载能耗
$\tilde{E}_{\mathrm{save}}$	加工机器执行开关机所节省的能耗

(转下页)

(接上页)

$\tilde{E}_{\text{off-on}}$	加工机器执行开关机所需能耗
$\tilde{wt}_{i,j}$	操作$o_{i,j}$完成后, 加工机器j的空载时间
$Z_{i,j}$	若$Z_{i,j}=1$, 加工机器关机; 否则, 保持开机
$\tilde{T}_{\text{B}}$	加工机器空载时, 执行开关机所需满足的最小时间
$WT_{\text{T},j}$	满足条件的$\widetilde{wt}_{i,j}$所构成的集合
$\widetilde{NLEC}_{\text{f}}$	工厂f的模糊总空载能耗
$\widetilde{LEC}$	加工阶段的模糊总带载能耗
$\tilde{t}_{\text{off}}^{\text{A}}$	装配机器关机所需时间
$\tilde{t}_{\text{on}}^{\text{A}}$	装配机器开机所需时间
T^{A}	装配机器可执行开关机的次数
$\tilde{S}_h^{\text{A}}$	装配产品P_h的开工时间
$\tilde{C}_h^{\text{A}}$	装配产品P_h的完工时间
$\varepsilon_{\text{idle}}^{\text{A}}$	装配机器单位时间的空载能耗
$\tilde{E}_{\text{save}}^{\text{A}}$	装配机器执行开关机所节省的能耗
$\tilde{E}_{\text{off-on}}^{\text{A}}$	装配机器执行开关机所需能耗
$\tilde{wt}_h^{\text{A}}$	产品P_h装配完成后, 装配机器的空载时间
Z_h^{A}	若$Z_h^{\text{A}}=1$, 装配机器关机; 否则, 保持开机
$\tilde{T}_{\text{B}}^{\text{A}}$	装配机器空载时, 执行开关机所需满足的最小时间
WT_{T}^{A}	满足条件的$\widetilde{wt}_h^{\text{A}}$所构成的集合
$\widetilde{NLEC}_{\text{A}}$	装配阶段的模糊总空载能耗
$\widetilde{LEC}_{\text{A}}$	装配阶段的模糊总带载能耗
$\widetilde{TNLEC}$	模糊总空载能耗
$\widetilde{TLEC}$	模糊总带载能耗
$\tilde{C}_{\max}$	模糊最大完工时间
$\widetilde{TEC}$	模糊总能耗

2.2 问题模型

FDAPFLSP分为两个阶段, 具体可描述为: 第1阶段为加工阶段, 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, 每台机器均可执行T次开关机, 关机所需时间为$\tilde{t}_{\text{off}}$, 开机所需时间为$\tilde{t}_{\text{on}}$, 执行一次开关机所需能耗为$\tilde{E}_{\text{off-on}}$; 第2阶段为装配阶段, 由一台装配机器将$N_h$个特定不同工件组装为对应产品$P_h$, 装配机器可执行$T^{\text{A}}$次开关机, 关机所需时间为$\tilde{t}_{\text{off}}^{\text{A}}$, 开机所需时间为$\tilde{t}_{\text{on}}^{\text{A}}$, 执行一次开关机所需能耗为$\tilde{E}_{\text{save}}^{\text{A}}$. 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$o_{i,j}$在加工过程中均不可中断, 且都具有相应的加工时间$\tilde{p}_{i,j}$, 构成产品P_h的所有工件均加工完成后才可进行装配, 且产品P_h具有相应的装配时间$\tilde{p}_h^{\text{A}}$.

FDAPFLSP的目标函数为模糊最大完工时间$\tilde{C}_{\max}$与模糊总能耗$\widetilde{TEC}$. 对该问题而言, $\widetilde{TEC}$由模糊总带载能耗$\widetilde{TLEC}$与模糊总空载能耗$\widetilde{TNLEC}$构成. $\widetilde{TLEC}$由加工阶段产生的带载能耗$\widetilde{LEC}$与装配阶段产生的带载能耗$\widetilde{LEC}_{\text{A}}$构成, 而$\widetilde{LEC}$仅与$\tilde{p}_{i,j}$有关, $\widetilde{LEC}_{\text{A}}$仅与$\tilde{p}_h^{\text{A}}$有关, 且FDAPFLSP中加工机器与装配机器仅考虑开关机, 均不涉及调速, 故对于同一问题的不同解, 其$\widetilde{TLEC}$均相同. 因此, 最小化$\widetilde{TEC}$可简化等价为最小化$\widetilde{TNLEC}$. 基于以上描述, 建立如下模型:

$$\tilde{C}_{\pi_1^{\text{f}},1}=\tilde{p}_{\pi_1^{\text{f}},1}, \tag{1}$$

$$\tilde{C}_{\pi_i^{\text{f}},1}=\tilde{C}_{\pi_{(i-1)}^{\text{f}},1}+\tilde{p}_{\pi_i^{\text{f}},1}, \tag{2}$$

$$\tilde{C}_{\pi_1^{\text{f}},j}=\tilde{C}_{\pi_1^{\text{f}},(j-1)}+\tilde{p}_{\pi_1^{\text{f}},j}, \tag{3}$$

$$\tilde{C}_{\pi_i^{\text{f}},j}=\tilde{C}_{\pi_{(i-1)}^{\text{f}},j}\vee\tilde{C}_{\pi_i^{\text{f}},(j-1)}+\tilde{p}_{\pi_i^{\text{f}},j}, \tag{4}$$

$$\tilde{C}_h=\max_{\pi_i^{\text{f}}\in P_h}\tilde{C}_{\pi_i^{\text{f}},M}, \tag{5}$$

$$\tilde{C}_h^{\text{A}}=\tilde{C}_{h-1}^{\text{A}}\vee\tilde{C}_h+\tilde{p}_h^{\text{A}}, \tag{6}$$

$$\tilde{C}_{\max}=\tilde{C}_{\pi_{\text{Q}}^{\text{A}}}^{\text{A}}, \tag{7}$$

$$T_{\text{B}}=(\tilde{E}_{\text{off-on}}/\varepsilon_{\text{idle}})\vee(\tilde{t}_{\text{off}}+\tilde{t}_{\text{on}}), \tag{8}$$

$$\begin{cases}\widetilde{wt}_{\pi_i^{\text{f}},j}=\tilde{S}_{\pi_{i+1}^{\text{f}},j}-\tilde{C}_{\pi_i^{\text{f}},j},\\ i\in\{1,2,\cdots,N_{\text{f}}-1\},\ j\in\{1,2,\cdots,M\},\end{cases} \tag{9}$$

$$\begin{cases}Z_{i,j}=\begin{cases}1,\ 若\widetilde{wt}_{\pi_i^{\text{f}},j}\geqslant\tilde{T}_{\text{B}}且\widetilde{wt}_{\pi_i^{\text{f}},j}\in WT_{\text{T},j},\\ 0,\ 否则,\end{cases}\\ i\in\{1,2,\cdots,N_{\text{f}}-1\},\ j\in\{1,2,\cdots,M\},\end{cases} \tag{10}$$

$$\begin{aligned}\widetilde{NLEC}_{\text{f}}=&\sum_{j=1}^{M}\varepsilon_{\text{idle}}\cdot(\tilde{C}_{N_{\text{f}},j}-\tilde{S}_{1,j})-\tilde{E}_{\text{save}}=\\&\sum_{j=1}^{M}\varepsilon_{\text{idle}}\cdot(\tilde{C}_{N_{\text{f}},j}-\tilde{S}_{1,j})-\\&\sum_{j=1}^{M}\sum_{i=1}^{N_{\text{f}}}(\varepsilon_{\text{idle}}\cdot\widetilde{wt}_{\pi_i^{\text{f}},j}-\tilde{E}_{\text{off-on}})\cdot Z_{i,j},\end{aligned} \tag{11}$$

$$T_{\text{B}}^{\text{A}}=(\tilde{E}_{\text{off-on}}^{\text{A}}/\tilde{\varepsilon}_{\text{idle}}^{\text{A}})\vee(\tilde{t}_{\text{off}}^{\text{A}}+\tilde{t}_{\text{on}}^{\text{A}}), \tag{12}$$

$$\widetilde{wt}_h^{\text{A}}=\tilde{S}_{h+1}^{\text{A}}-\tilde{C}_h^{\text{A}},\ h\in\{1,2,\cdots,Q-1\}, \tag{13}$$

$$\begin{cases}Z_h^{\text{A}}=\begin{cases}1,\ 若\widetilde{wt}_h^{\text{A}}\geqslant\tilde{T}_{\text{B}}^{\text{A}}且\widetilde{wt}_h^{\text{A}}\in WT_{T_{\text{A}}}^{\text{A}},\\ 0,\ 否则,\end{cases}\\ h\in\{1,2,\cdots,Q-1\},\end{cases} \tag{14}$$

$$\begin{aligned}\widetilde{NLEC}_{\text{A}}=&\varepsilon_{\text{idle}}^{\text{A}}\cdot(\tilde{C}_{\text{Q}}^{\text{A}}-\tilde{S}_1^{\text{A}})-\tilde{E}_{\text{save}}^{\text{A}}=\\&\varepsilon_{\text{idle}}^{\text{A}}\cdot(\tilde{C}_{\text{Q}}^{\text{A}}-\tilde{S}_1^{\text{A}})-\\&\sum_{h=1}^{Q-1}(\varepsilon_{\text{idle}}^{\text{A}}\cdot\widetilde{wt}_h^{\text{A}}-\tilde{E}_{\text{off-on}}^{\text{A}})\cdot Z_h^{\text{A}},\end{aligned} \tag{15}$$

$$\widetilde{TNLEC}=\sum_{f=1}^{F}\widetilde{NLEC}_{\text{f}}+\widetilde{NLEC}_A, \tag{16}$$

$$\widetilde{TEC}=\widetilde{TLEC}+\widetilde{TNLEC}, \tag{17}$$

其中: 式(1)–(7)为模糊最大完工时间$\tilde{C}_{\max}$的计算公

式; 式(8)–(11)为加工阶段每个工厂所产生空载能耗$\widetilde{NLEC}_{\rm f}$的计算公式; 式(12)–(15)为装配阶段所产生空载能耗$\widetilde{NLEC}_{\rm A}$的计算公式; 式(16)为模糊总空载能耗$T\widetilde{NLEC}$的计算公式; 式(17)为模糊总能耗$\widetilde{TEC}$的计算公式.

2.3　模糊数

2.3.1　模糊数的运算

在计算FDAPFLSP不同个体或解对应的目标函数值时, 需对模糊数(即加工和装配时间)进行加、减、乘、取大等运算. 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(triangular fuzzy number, TFN), 定义其加法运算如式(18)所示:

$$\tilde{U}+\tilde{W}=(u_1+w_1,u_2+w_2,u_3+w_3). \tag{18}$$

定义其减法运算如式(19)所示:

$$\tilde{U}-\tilde{W}=(u_1-w_1,u_2-w_2,u_3-w_3). \tag{19}$$

ζ为任意实数, 定义其乘法运算如式(20)所示:

$$\zeta\cdot\tilde{U}=(\zeta\cdot u_1,\zeta\cdot u_2,\zeta\cdot u_3). \tag{20}$$

取大、比较运算均由模糊数排序实现. Sakawa等[5]提出且在近年调度文献中广泛使用的排序准则具体定义如下:

1) 计算$L_1(\tilde{X})=\dfrac{x_1+2\cdot x_2+x_3}{4}$, 并将$L_1$作为排序的首要依据;

2) 若两个TFN的L_1相等, 定义$L_2(\tilde{X})=x_2$作为排序的次要依据;

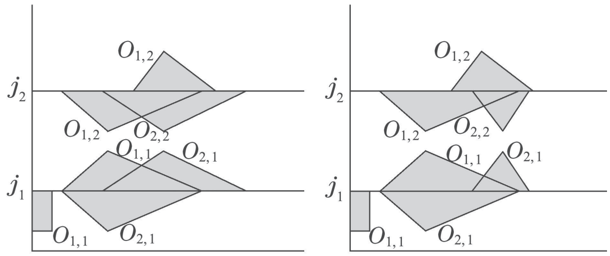
3) 若两个TFN的L_1和L_2都相等, 定义$L_3(\tilde{X})=x_3-x_1$作为排序的第3条依据.

进而, 两个三角模糊数的取大运算定义为: 若$\tilde{U}>\tilde{W}$, 则$\tilde{U}\vee\tilde{W}=\tilde{U}$; 否则, $\tilde{U}\vee\tilde{W}=\tilde{W}$.

2.3.2　模糊数排序修正准则

通过第2.3.1节的描述可知, Sakawa等[5]所提排序准则是对两个TFN进行近似精确化处理后, 通过对其近似精确值进行比较后, 确定两个TFN之间的大小关系. Li等[7]所提排序准则也是对TFN进行近似精确化处理后再进行比较.

但若将此类排序准则直接用于生产调度问题, 可能会出现同一时刻同一台机器上安排两个工件进行加工(即违反问题约束)的情况, 譬如, 使用Sakawa等[5]所提排序准则对$O_{1,2}$和$O_{2,1}$的完工时间进行比较, 得到$O_{2,2}$的开工时间, 此时$O_{2,2}$开工时间的最乐观值小于$O_{1,2}$完工时间的最乐观值, 如图1(a)所示; $O_{2,2}$开工时间的最悲观值小于$O_{1,2}$完工时间的最悲观值, 如图1(b)所示. 图1中下方为操作的模糊开工时间, 上方为操作的模糊完工时间. 此类情况的出现, 使得该准则在实际生产过程中不可行.



(a) 情况1　　(b) 情况2

图 1 违反问题约束情况示意图

Fig. 1 Illustrations of two cases that constrains are violated

对此, 提出一种适用于生产调度问题的模糊数修正排序准则. 记机器j上工件的加工序列为π^j.

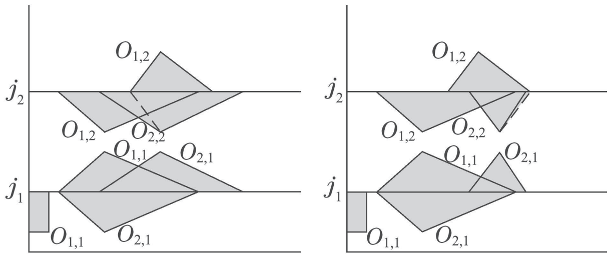
首先使用Sakawa等[5]所提排序准则, 比较后得到机器j上加工第i工件($i>1$) 的操作$O_{\pi_i^j,j}$的理论开工时间(w_1,w_2,w_3); 然后将$O_{\pi_{i-1}^j,j}$的完工时间(u_1,u_2,u_3)与$O_{\pi_i^j,j}$的理论开工时间(w_1,w_2,w_3)进行比较、修正处理, 得到$O_{\pi_i^j,j}$的实际开工时间.

1) 若$u_1>w_1$且$u_3\leqslant w_3$, 则$O_{\pi_i^j,j}$的实际开工时间为(u_1,w_2,w_3);

2) 若$u_1\leqslant w_1$且$u_3>w_3$, 则$O_{\pi_i^j,j}$的实际开工时间为(w_1,w_2,u_3);

3) 若$u_1>w_1$且$u_3>w_3$, 则$O_{\pi_i^j,j}$的实际开工时间为(u_1,w_2,u_3).

图1所举示例经修正准则处理后如图2所示, 其中: $O_{2,2}$的理论开工时间用实线表示, 经修正准则处理后所得实际开工时间用虚线表示.



(a) 情况1　　(b) 情况2

图 2 修正准则处理后情况示意图

Fig. 2 Illustrations of two cases that correction rule is used

3　混合交叉熵算法

3.1　编码与解码

个体的编码由产品装配序列$\pi^{\rm A}$和工件加工序列π两部分组成. 其中, 属于同一产品的工件在π中相邻放置. 个体编码示意图如图3所示.

3	2	1	4

(a) 产品装配序列$\pi^{\rm A}$

3	1	2	5	8	7	4	6	9
$h=3$	$h=3$	$h=2$	$h=2$	$h=1$	$h=1$	$h=4$	$h=4$	$h=4$

(b) 工件加工序列π

图 3 个体编码示意图

Fig. 3 Illustration of individual coding

对于每个个体, 均对其工件加工序列π采用文献[3]中所提NR2规则进行解码. 例如, 对于

$$F=2,\ N=9,\ M=2,\ Q=4,$$

编码如图3所示的问题. 各阶段数据如表2所示. 解码后所得甘特图如图4所示.

表 2 各阶段数据

Table 2 Data of each stage

		产品1		产品2		产品3		产品4		
模糊装配时间		(13,15,17)		(10,13,14)		(14,16,17)		(8,9,11)		
所包含工件		7	8	2	5	1	3	4	6	9
模糊加工时间	机器1	(3,6,9)	(6,9,12)	(7,10,11)	(6,8,10)	(5,9,13)	(2,5,9)	(2,5,6)	(2,4,7)	(1,2,3)
	机器2	(2,5,11)	(5,6,8)	(3,5,8)	(4,8,11)	(3,7,10)	(7,11,15)	(7,8,11)	(1,2,3)	(5,9,11)

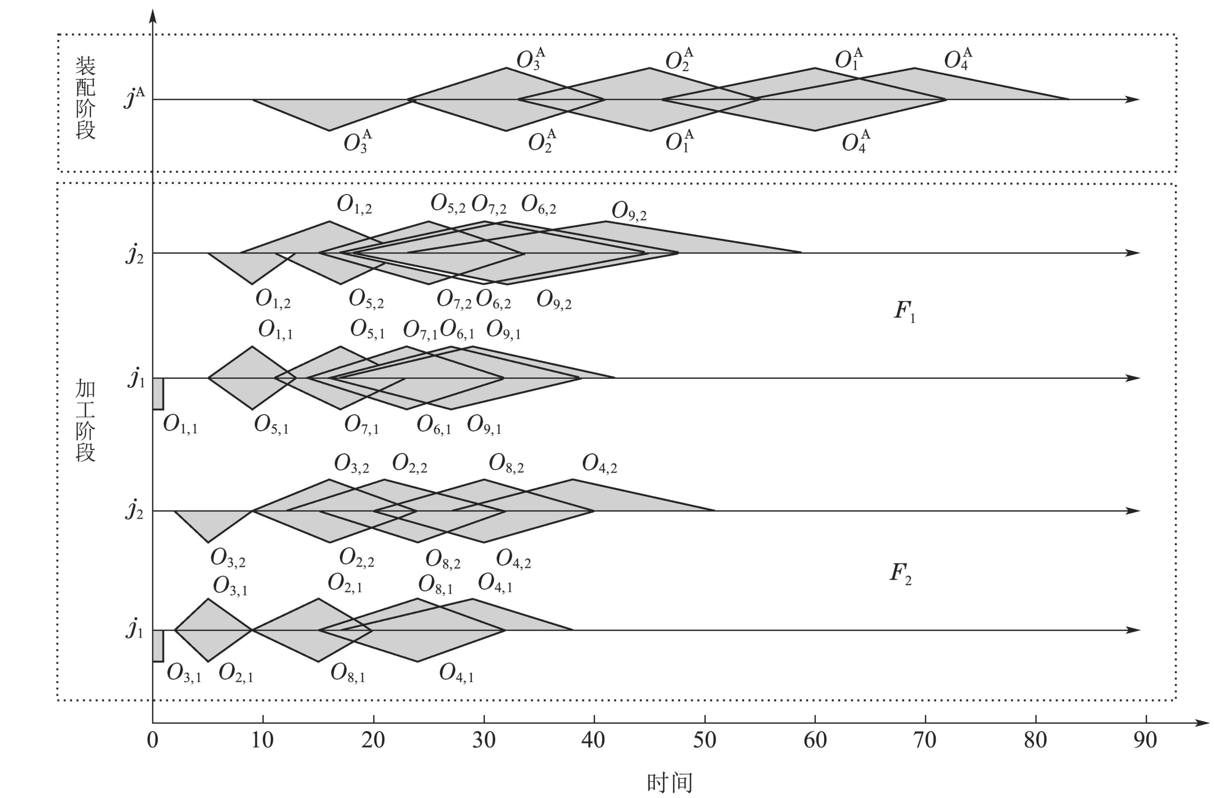


图 4 解码后所得甘特图

Fig. 4 Gantt chart obtained after decoded

3.2 种群初始化

本文采用随机初始化方法生成种群, 以保证种群的多样性与分散性.

3.3 交叉熵算法

使用CEA对种群进行全局搜索. 首先, 使用概率分布矩阵Λ表示ρ个优质个体所包含信息, 矩阵元素$\zeta_{i,j}$表示工件i出现在π中第j位的概率, 矩阵Λ如式(21)所示:

$$\Lambda=\begin{bmatrix} \xi_{1,1} & \cdots & \xi_{1,j} & \cdots & \xi_{1,N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \xi_{i,1} & \cdots & \xi_{i,j} & \cdots & \xi_{i,N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \xi_{N,1} & \cdots & \xi_{N,j} & \cdots & \xi_{N,N} \end{bmatrix}, \tag{21}$$

其中$\zeta_{i,j}\in[0,1]$且$\sum\limits_{i=1}^{N}\zeta_{i,j}=1,\ j=1,2,\cdots,N.$

初始矩阵$\Lambda^{(0)}$采用均匀分布, 即$\Lambda_{ij}^{(0)}=\dfrac{1}{N}$, 概率分布按式(22)进行更新:

$$\Lambda_{ij}^{(\nu)}=\eta\cdot\frac{1}{\rho}\sum_{l=1}^{\rho}I_{\{a_{ij=1}\}}^{l}+(1-\eta)\cdot\Lambda_{ij}^{(\nu-1)},\quad(22)$$

其中: k为迭代次数, $\eta\in(0,1)$为学习率, $I_{\{a_{ij=1}\}}^{l}$为指示函数, ρ为优质个体数量.

其次, 按照如下步骤逐位对概率分布矩阵Λ进行采样以生成新个体:

步骤 1　若π^{A}中当前已确定产品位置的个数ϖ等于总产品数Q, 转至步骤2; 否则, 转至步骤3.

步骤 2　记π^{A}第ϖ位所对应产品为P_{π_ϖ}, 若π中当前已确定P_{π_ϖ}包含工件的位置的个数ς_{π_ϖ}小于P_{π_ϖ}所包含的总工件数N_{π_ϖ}, 转至步骤4; ς_{π_ϖ}等于N_{π_ϖ}, 终止循环.

步骤 3　若$\varpi=0$, 转至步骤5. 否则, 记π^{A}第ϖ位所对应产品为P_{π_ϖ}, 若π中当前已确定P_{π_ϖ}包含工件的位置的个数ς_{π_ϖ}小于P_{π_ϖ}所包含的总工件数N_{π_ϖ}, 转至步骤4; 若ς_{π_ϖ}等于N_{π_ϖ}, 转至步骤5.

步骤 4　从P_{π_ϖ}的剩余可选工件中进行采样, 确定出现在π中当前位置(当$\varpi=1$时, 为第$\varsigma_{\pi_\varpi}+1$位; 当$\varpi>1$时, 为第$\sum\limits_{i=1}^{\varpi-1}N_{\pi_i}+\varsigma_{\pi_\varpi}+1$位)的工件; 若剩余可选工件出现在该位置的概率均为0, 则随机从中选择一个置于该位置, 转至步骤1.

步骤 5　从可选工件中进行采样, 确定出现在π中当前位置(当$\varpi=0$时, 为第1位; 当$\varpi\neq0$时, 为第$\sum\limits_{i=1}^{\varpi}N_{\pi_i}+1$位)的工件; 若可选工件出现在该位置的概率均为0, 则随机从中选择一个置于该位置. 且将该工件所对应产品置于π^{A}中第$\varpi+1$位, 转至步骤1.

3.4　自适应变邻域局部搜索

对于多目标组合优化问题, 每个最优非劣解集中的解(简称最优非劣解)在所有邻域结构下均为局部最优非劣解, 而接近最优非劣解的优质非劣解往往在多种邻域结构下均为局部最优非劣解. 采用单一邻域的迭代搜索容易较早达到并陷入该邻域结构的局部最优非劣解(可能有多个), 但该非劣解的质量大都一般. 相对于常规的流水线调度问题, 本文研究的FDAPFLSP中约束和决策变量明显变多. 这使得本文问题的解空间更加庞大和不规则, 大量优质局部最优非劣解不规则地分散在FDAPFLSP 的解空间内, 增加了算法获取优质非劣解集的难度. 因此, 本节设计基于多种邻域操作的自适应变邻域局部搜索, 在算法进化过程中根据各邻域操作的贡献率动态选择合适的邻域操作执行搜索, 可在算法到达多种邻域结构共同的局部最优非劣解前以较高效率一直持续逼近最优非劣解集, 从而增强搜索的深度, 有利于算法获得优质非劣解集. 具体来说, 本节采用交换、插入、逆序操作(即求解调度问题常用的有效邻域操作)设计基于产品序列与基于工件序列的两类自适应局部搜索, 用于对CEA发现的解空间优质区域进行深入搜索.

3.4.1　基于产品序列的自适应局部搜索

设计了6种作用于产品序列π^{A}的邻域操作. π^{A}中的任意产品调整, 工件序列π中对应部分工件整体进行调整.

每种操作的初始选择概率相同, 当一次迭代完成(即种群中全部个体均进行了邻域操作)后, 其选择概率$se_{\mathrm{LS}(g)}$的更新步骤如下:

步骤 1　按式(23)分别计算每种邻域操作的贡献率$e_{\mathrm{LS}(g)}$:

$$e_{\mathrm{LS}(g)}=ND_{\mathrm{LS}(g)}/NG_{\mathrm{LS}(g)},\quad(23)$$

其中: $NG_{\mathrm{LS}(g)}$为一次迭代完成后, 邻域操作g所执行的总次数; $ND_{\mathrm{LS}(g)}$为一次迭代完成后, 邻域操作g对个体进行更新后所得新解不被更新前的旧解支配的次数.

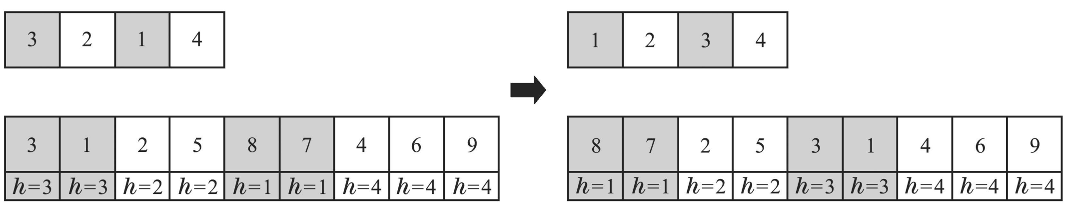
步骤 2　计算$\sum\limits_{g=1}^{6}e_{\mathrm{LS}(g)}$, 若$\sum\limits_{g=1}^{6}e_{\mathrm{LS}(g)}=0$, 则不对邻域操作的选择概率$se_{\mathrm{LS}(g)}$进行更新, 并终止循环; 否则转到步骤3.

步骤 3　按式(24)分别计算更新每种邻域操作的选择概率$se_{\mathrm{LS}(g)}$:

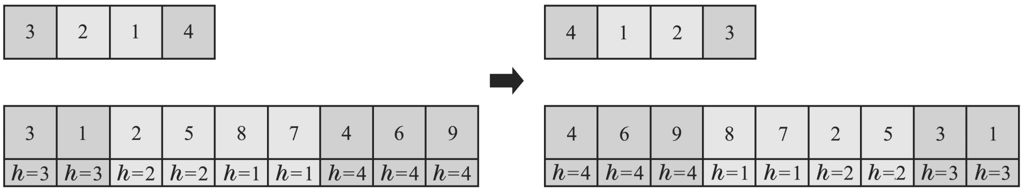
$$se_{\mathrm{LS}(g)}=e_{\mathrm{LS}(g)}/\sum_{g=1}^{6}e_{\mathrm{LS}(g)}.\quad(24)$$

6种邻域操作分别如下所示:

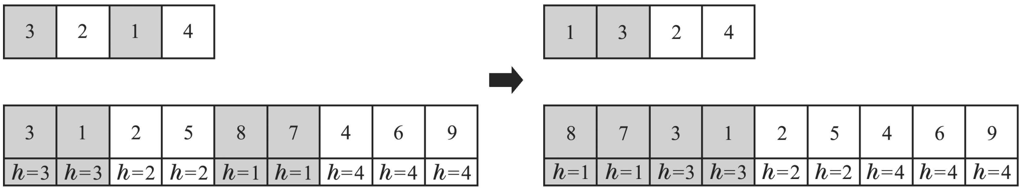
1) 产品序列交叉操作, 从产品序列中随机选择两位并进行交换.



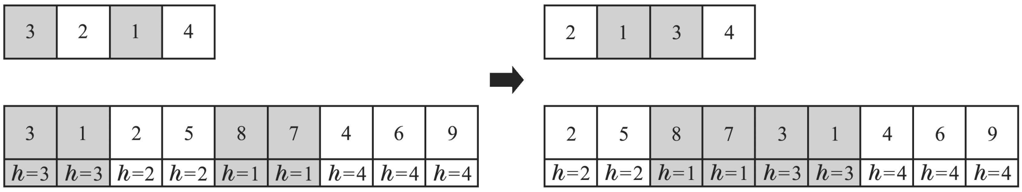
2) 产品序列逆序操作, 从产品序列中随机选择两位, 将包含所选两位及其之间的部分进行逆序排列.



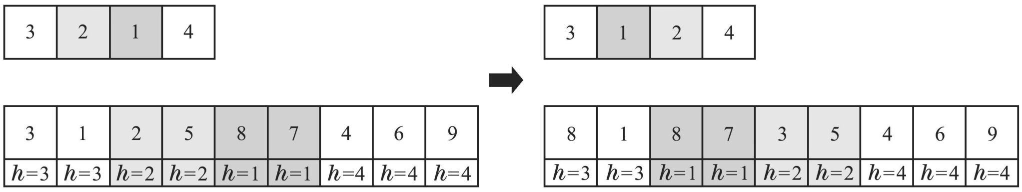
3) 产品序列前向插入操作, 从产品序列中随机选择两位, 位置编号大的产品插入到位置编号小的产品之前.



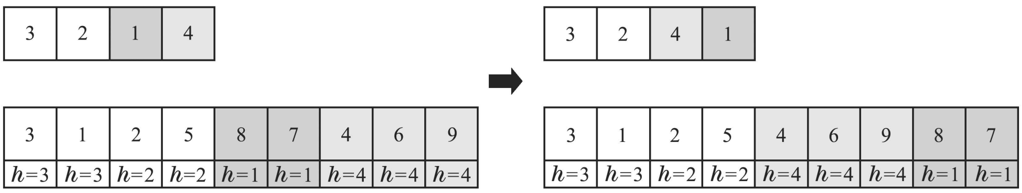
4) 产品序列后向插入操作, 从产品序列中随机选择两位, 位置编号小的产品插入到位置编号大的产品之后.



5) 产品序列前向相邻交叉操作, 从产品序列中随机选择一位, 向前和相邻产品交换位置.



6) 产品序列后向相邻交叉操作, 从产品序列中随机选择一位, 向后和相邻产品交换位置.



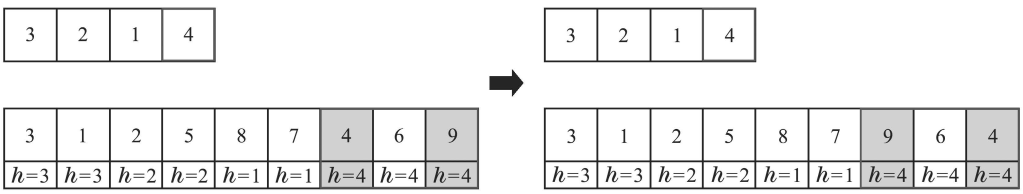
3.4.2 基于工件序列的自适应局部搜索

设计了6种作用于产品P_h的工件加工序列π_h^{A}的邻域操作. 进行此类操作时, 需要先确定产品号, 后对其所对应的π_h^{A}中工件进行调整.

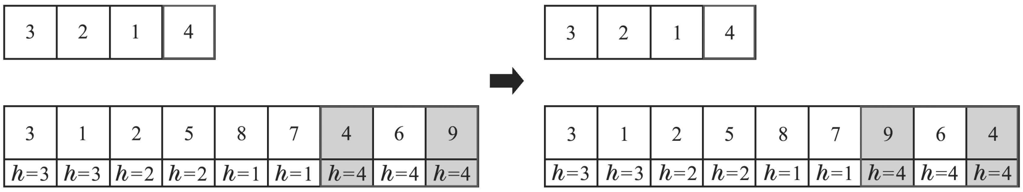
每种操作的初始选择概率相同, 随着迭代过程的进行, 其选择的概率$se_{\mathrm{LS}(g)}$的更新步骤如第2.4.1节所示.

6种邻域操作分别如下所示:

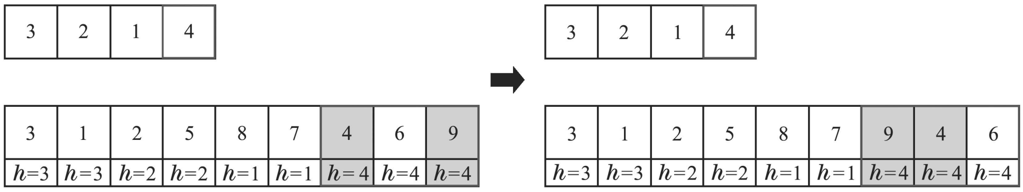
1) 工件序列交叉操作, 从产品P_h的工件加工序列π_h^{A}中随机选择两位并进行交换.



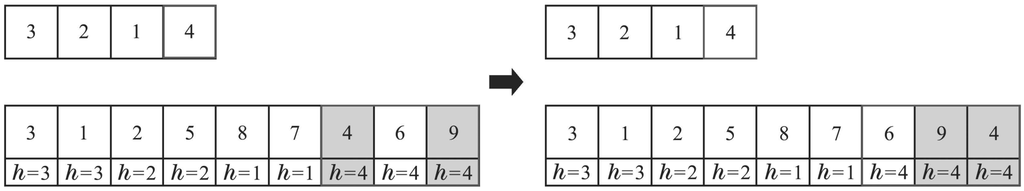
2) 工件序列逆序操作, 从产品P_h的工件加工序列π_h^{A}中随机选择两位, 将包含所选两位及其之间的部分进行逆序排列.



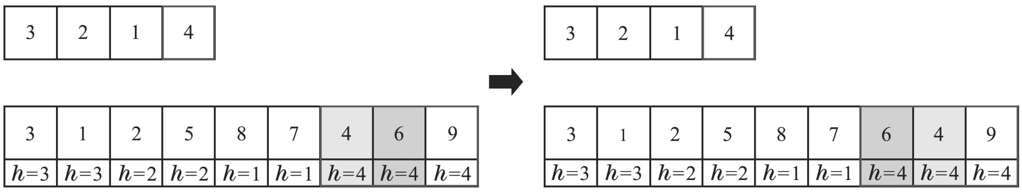
3) 工件序列前向插入操作, 从产品P_h的工件加工序列π_h^{A}中随机选择两位, 位置编号大的工件插入到位置编号小的工件之前.



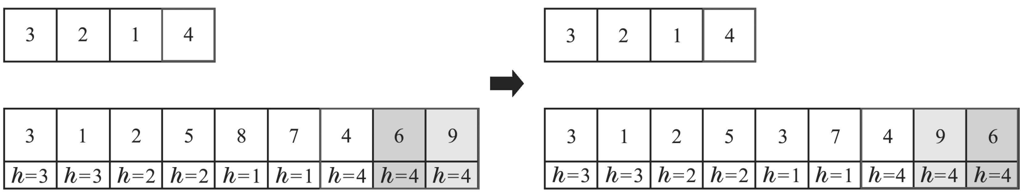
4) 工件序列后向插入操作, 从产品P_h的工件加工序列π_h^{A}中随机选择两位, 位置编号小的工件插入到位置编号大的工件之后.



5) 工件序列前向相邻交叉操作, 从产品P_h的工件加工序列π_h^{A}中随机选择一位, 向前和相邻工件交换位置.



6) 工件序列后向相邻交叉操作, 从产品P_h的工件加工序列π_h^{A}中随机选择一位, 向后和相邻工件交换位置.



3.5　算法流程

根据上述算法描述, 整个算法流程描述如下:

步骤 1　初始化种群.

步骤 2　对种群中个体进行评价.

步骤 3　对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. 对其排序后, 选择前$\mathrm{popsize} \times \theta_\alpha$个精英个体直接保留至新种群.

步骤 4　选择当代种群中前$\mathrm{popsize} \times \theta_\alpha$个个体更新交叉熵算法的概率模型, 并生成新种群中剩余$\mathrm{popsize} \times (1-\theta_\alpha)$个个体.

步骤 5　对新种群中每个个体, 分别依据6种基于产品序列的邻域操作的选择概率, 随机选择一种操作对其进行更新. 个体更新完成后判断个体是否改善, 若改善, 则接受新解; 否则, 有一定的概率接受新解. 全部个体完成操作后, 更新基于产品序列的自适应局部搜索的选择概率.

步骤6 对新种群中每个个体, 分别依据6种基于工件序列的邻域操作的选择概率, 随机选择一种操作对其进行更新. 个体更新完成后判断个体是否改善, 若改善, 则接受新解; 否则, 有一定的概率接受新解. 全部个体完成操作后, 更新基于工件序列的自适应局部搜索的选择概率.

步骤7 使用所得非支配解集更新Pareto档案集.

步骤8 判断是否满足终止条件, 若不满足则转到步骤2; 否则, 终止循环.

4 实验分析与设计

由于目前没有适合FDAPFLSP的标准算例, 本文所有测试问题均在由Hatami等[3]为解决DAPFSP所提供算例的基础上生成, 每个测试问题均包含工件的模糊加工时间、产品的模糊装配时间与产品约束3部分.

4.1 参数设置

FDAPFLSP中参数取值如下: 加工机器单位时间的空载能耗$\varepsilon_{\rm idle}=2$、加工机器执行开关机所需能耗$\tilde{E}_{\rm off-on}=(8,8,8)$、加工机器执行一次开关机所需时间$\tilde{t}_{\rm off}+\tilde{t}_{\rm on}=(5,5,5)$、装配机器单位时间的空载能耗$\varepsilon^{\rm A}_{\rm idle}=3$、装配机器执行开关机所需能耗$\tilde{E}^{\rm A}_{\rm off-on}=(20,20,20)$、装配机器执行一次开关机所需时间$\tilde{t}^{\rm A}_{\rm off}+\tilde{t}^{\rm A}_{\rm on}=(9,9,9)$; 加工机器的可执行开关机次数$T$可通过$T=\lceil[(N/F)-2]/5\rceil$ 计算获取, 装配机器的可执行开关机次数$T^{\rm A}$可通过$T^{\rm A}=\lceil(Q-2)/5\rceil$计算获取.

HCEA中的关键参数为种群容量popsize, 学习率η, 精英个体所占比例θ_α, 统计样本的ρ分位数与种群容量的比值θ_ρ, 进行基于产品序列的局部操作后差解的接受概率$\varphi_{\rm P}$, 进行基于工件序列的局部操作后差解的接受概率$\varphi_{\rm J}$. 对于小规模问题, HCEA的参数设置如下: popsize $=100$, $\eta=0.3$, $\theta_\alpha=0.2$, $\theta_\rho=0.3$, $\varphi_{\rm P}=0.07$, $\varphi_{\rm J}=0.07$; 对于大规模问题, HCEA的参数设置如下: popsize $=100$, $\eta=0.3$, $\theta_\alpha=0.3$, $\theta_\rho=0.1$, $\varphi_{\rm P}=0.07$, $\varphi_{\rm J}=0.03$.

4.2 仿真结果与比较

为验证HCEA求解FDAPFLSP的有效性, 本节将HCEA与SPEA2[24], IMMOGLS[25], MOBSO[16] 进行比较. SPEA2与IMMOGLS是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低碳调度问题的有效算法. 各算法均采用本文提出的模糊数排序修正准则.

对于不同规模问题, 设定每种算法的运行时间均为$N\times M\times 20$ ms. 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R_N如式(25)所示, 衡量的是算法所得S_j的整体质量; 第2个指标N_N如式(26)所示, 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.

$$R\_N(S_j)=\frac{|S_j-\{x\in S_j|\exists y\in S:y\prec x\}|}{|S_j|},\quad(25)$$

$$N\_N(S_j)=|S_j-\{x\in S_j|\exists y\in S:y\prec x\}|,\quad(26)$$

其中: S表示所有算法所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.

对于小规模问题, 各算法比较结果如表3所示. 对于大规模问题, 各算法比较结果如表4所示. 由表3–4可知, HCEA在绝大部分问题上的测试结果都明显优于比较算法, 这表明HCEA是求解FDAPFLSP的有效算法.

在比较算法中, SPEA2, IMMOGLS和MOBSO均采用交叉、变异等传统操作生成新种群以实现全局搜索, 同时SPEA2无局部搜索, IMMOGLS在各目标函数随机加权和的方向上执行基于常规插入邻域的局部搜索, MOBSO执行基于常规插入、交换邻域的局部搜索. HCEA通过采样CEA概率模型生成新种群来实现全局搜索, 可在一定程度上避免传统交叉、变异等操作存在的对较优解优良模式破坏的问题[26], 从而能更好地保留较优解信息并合理引导搜索方向. 同时, HCEA的局部搜索将更多的常规邻域操作(即插入、逆序、交换)进一步细分为12种基于产品或工件的邻域操作, 并依据每种邻域操作在搜索过程中的效果动态择优确定应执行的邻域搜索, 可实现对邻域搜索行为更精细的控制, 从而能进行更深入和更高效率的搜索. 由于HCEA具有这些优势, 故在上述实验中取得整体最好的成绩.

5 结论

本文结合模糊数建立模糊分布式装配流水线低碳调度问题(fuzzy distributed assembly permutation flow-shop low-carbon scheduling problem, FDAPFLSP)的模型, 并提出一种混合交叉熵算法(hybrid cross-entropy algorithm, HCEA) 进行求解. 主要贡献包括: 1) 采用模糊数表示加工和装配时间, 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, 首次建立FDAPFLSP的问题模型; 2) 在已有排序准则的基础上, 设计一种三角模糊数排序修正准则, 以确保由准则获得的解(即调度方案)能更好满足机器在同一时刻只能加工一个工件的基本约束, 从而提升准则的实用性; 3) 在采用CEA执行全局搜索的基础上, 设计基于邻域操作贡献率的变邻域局部搜索对全局搜索发现的优质区域执行深入、细致的搜索, 可增强算法性能. 未来研究将把HCEA扩展用于求解不确定车辆配送问题.

表 3 小规模问题算法比较结果

Table 3 Comparisons of algorithm for small scale problem

$N \times M \times F \times Q$	SPEA2		IMMOGLS		MOBSO		HCEA		$N \times M \times F \times Q$	SPEA2		IMMOGLS		MOBSO		HCEA	
	R_N	N_N	R_N	N_N	R_N	N_N	R_N	N_N		R_N	N_N	R_N	N_N	R_N	N_N	R_N	N_N
$16 \times 2 \times 2 \times 3$	0.010	0.05	0.611	4.70	0.392	2.10	**0.826**	**8.00**	$24 \times 2 \times 2 \times 3$	0.005	0.05	**0.586**	5.25	0.224	1.20	0.498	**5.50**
$16 \times 2 \times 2 \times 4$	0.000	0.00	0.196	0.50	0.025	0.05	**0.858**	**2.70**	$24 \times 2 \times 2 \times 4$	0.000	0.00	0.503	3.95	0.189	1.15	**0.636**	**5.55**
$16 \times 2 \times 3 \times 3$	0.006	0.05	**0.654**	**5.05**	0.148	1.00	0.553	4.15	$24 \times 2 \times 3 \times 3$	0.006	0.05	0.395	3.30	0.119	0.55	**0.612**	**6.25**
$16 \times 2 \times 3 \times 4$	0.000	0.00	0.000	0.00	0.000	0.00	**1.000**	**2.00**	$24 \times 2 \times 3 \times 4$	0.015	0.10	0.389	2.70	0.123	0.80	**0.658**	**5.35**
$16 \times 2 \times 4 \times 3$	0.083	0.25	0.500	1.55	0.092	0.25	**0.942**	**2.95**	$24 \times 2 \times 4 \times 3$	0.000	0.00	0.376	2.50	0.108	0.55	**0.701**	**5.45**
$16 \times 2 \times 4 \times 4$	0.000	0.00	**0.700**	**0.90**	0.000	0.00	0.350	0.40	$24 \times 2 \times 4 \times 4$	0.021	0.10	0.240	1.00	0.017	0.10	**0.772**	**3.50**
$16 \times 3 \times 2 \times 3$	0.000	0.00	0.093	0.30	0.000	0.00	**0.942**	**1.95**	$24 \times 3 \times 2 \times 3$	0.010	0.05	0.397	2.45	0.148	0.80	**0.818**	**5.70**
$16 \times 3 \times 2 \times 4$	0.000	0.00	0.218	0.90	0.085	0.30	**0.825**	**3.55**	$24 \times 3 \times 2 \times 4$	0.005	0.05	**0.560**	5.65	0.166	1.55	0.527	**6.60**
$16 \times 3 \times 3 \times 3$	0.023	0.10	0.392	1.70	0.100	0.40	**0.858**	**3.70**	$24 \times 3 \times 3 \times 3$	0.017	0.05	0.153	0.75	0.033	0.20	**0.983**	**2.60**
$16 \times 3 \times 3 \times 4$	0.000	0.00	0.347	1.60	0.060	0.20	**0.663**	**3.05**	$24 \times 3 \times 3 \times 4$	0.010	0.05	0.381	2.00	0.011	0.10	**0.739**	**4.45**
$16 \times 3 \times 4 \times 3$	0.020	0.10	0.520	2.80	0.260	1.00	**0.800**	**4.00**	$24 \times 3 \times 4 \times 3$	0.007	0.05	0.404	3.60	0.148	1.45	**0.605**	**5.90**
$16 \times 3 \times 4 \times 4$	0.017	0.05	0.250	0.60	0.000	0.00	**0.858**	**2.10**	$24 \times 3 \times 4 \times 4$	0.006	0.05	0.395	2.30	0.043	0.30	**0.628**	**4.25**

表 4 大规模问题算法比较结果

Table 4 Comparisons of algorithm for large scale problem

$N \times M \times F \times Q$	SPEA2		IMMOGLS		MOBSO		HCEA		$N \times M \times F \times Q$	SPEA2		IMMOGLS		MOBSO		HCEA	
	R_N	N_N	R_N	N_N	R_N	N_N	R_N	N_N		R_N	N_N	R_N	N_N	R_N	N_N	R_N	N_N
$100 \times 5 \times 4 \times 30$	0.113	0.15	0.188	0.90	0.025	0.15	**0.881**	**3.30**	$200 \times 5 \times 4 \times 30$	0.222	0.55	0.117	0.40	0.010	0.05	**0.811**	**3.02**
$100 \times 5 \times 4 \times 40$	0.238	0.40	0.060	0.25	0.006	0.05	**0.763**	**1.70**	$200 \times 5 \times 4 \times 40$	0.250	0.70	0.181	0.70	0.015	0.10	**0.892**	**4.30**
$100 \times 5 \times 6 \times 30$	0.101	0.35	0.180	0.65	0.015	0.10	**0.933**	**2.75**	$200 \times 5 \times 6 \times 30$	0.273	0.65	0.137	0.50	0.032	0.15	**0.703**	**2.40**
$100 \times 5 \times 6 \times 40$	0.134	0.40	0.199	1.10	0.055	0.40	**0.836**	**3.95**	$200 \times 5 \times 6 \times 40$	0.290	0.85	0.234	1.05	0.056	0.30	**0.786**	**3.90**
$100 \times 5 \times 8 \times 30$	0.038	0.10	0.149	0.60	0.028	0.15	**0.886**	**3.00**	$200 \times 5 \times 8 \times 30$	0.230	0.50	0.342	1.25	0.054	0.30	**0.768**	**3.60**
$100 \times 5 \times 8 \times 40$	0.126	0.30	0.198	0.95	0.015	0.10	**0.910**	**3.45**	$200 \times 5 \times 8 \times 40$	0.306	1.05	0.208	1.15	0.101	0.70	**0.729**	**4.20**
$100 \times 10 \times 4 \times 30$	0.300	0.40	0.145	0.60	0.010	0.05	**0.790**	**2.80**	$200 \times 10 \times 4 \times 30$	0.226	0.70	0.168	0.90	0.056	0.35	**0.845**	**4.50**
$100 \times 10 \times 4 \times 40$	0.358	0.55	0.074	0.35	0.017	0.10	**0.796**	**2.35**	$200 \times 10 \times 4 \times 40$	0.396	0.70	0.144	0.45	0.018	0.10	**0.640**	**2.50**
$100 \times 10 \times 6 \times 30$	0.105	0.30	0.202	0.95	0.035	0.20	**0.846**	**3.95**	$200 \times 10 \times 6 \times 30$	0.295	0.85	0.079	0.40	0.000	0.00	**0.715**	**2.85**
$100 \times 10 \times 6 \times 40$	0.049	0.20	0.115	0.45	0.035	0.20	**0.940**	**3.85**	$200 \times 10 \times 6 \times 40$	0.264	0.55	0.213	0.70	0.042	0.20	**0.684**	**2.55**
$100 \times 10 \times 8 \times 30$	0.073	0.20	0.154	0.75	0.024	0.15	**0.915**	**4.25**	$200 \times 10 \times 8 \times 30$	0.429	1.10	0.220	0.70	0.078	0.50	**0.583**	**2.45**
$100 \times 10 \times 8 \times 40$	0.229	0.55	0.226	0.95	0.036	0.25	**0.806**	**3.55**	$200 \times 10 \times 8 \times 40$	0.324	0.65	0.230	0.70	0.028	0.20	**0.734**	**2.55**

参考文献:

[1] AI Ziyi, LEI Deming. A novel shufflfled frog leaping algorithm for low carbon flflexible job shop scheduling. *Control Theory & Applications*, 2017, 34(10): 1361 – 1368.
(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. 控制理论与应用, 2017, 34(10): 1361 – 1368.)

[2] LI Ming, LEI Deming. Novel imperialist competitive algorithm for many-objective flexible job shop scheduling. *Control Theory & Applications*, 2019, 36(6): 893 – 901.
(李明, 雷德明. 基于新型帝国竞争算法的高维多目标柔性作业车间调度. 控制理论与应用, 2019, 36(6): 893 – 901.)

[3] HATAMI S, RUIZ R, ANDRÉS-ROMANO C. The distributed assembly permutation flowshop scheduling problem. *International Journal of Production Research*, 2013, 51(17): 5292 – 5308.

[4] SAKAWA M, MORI T. An efficient genetic algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy duedate. *Computers & Industrial Engineering*, 1999, 36(2): 325 – 341.

[5] SAKAWA M, KUBOTA R. Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms. *European Journal of Operational Research*, 2000, 120(2): 393 – 407.

[6] LEI D M. Fuzzy job shop scheduling problem with availability constraints. *Computers & Industrial Engineering*, 2010, 58(4): 610 – 617.

[7] LI Shanghan, HU Rong, QIAN Bin, et al. Hyper-heuristic genetic algorithm for solving fuzzy flexible job shop scheduling problem. *Control Theory & Application*, 2020, 37(2): 316 – 330.
(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. 控制理论与应用, 2020, 37(2): 316 – 330.)

[8] LIN J, WANG Z J, LI X D. A backtracking search hyper-heuristic for the distributed assembly flow-shop scheduling problem. *Swarm and Evolutionary Computation*, 2017, 36: 124 – 135.

[9] SANG H Y, PAN Q K, LI J Q, et al. Effective invasive weed optimization algorithms for distributed assembly permutation flowshop problem with total flowtime criterion. *Swarm and Evolutionary Computation*, 2019, 44: 64 – 73.

[10] FERONE D, HATAMI S, GONZÁLEZ NEIRA E M, et al. A biased-randomized iterated local search for the distributed assembly permutation flow-shop problem. *International Transactions in Operational Research*, 2020, 27(3): 1368 – 1391.

[11] GU Xingsheng. A survey of production scheduling under uncertainly. *Journal of East China University of Science and Technology*, 2000, 26(5): 441 – 446.
(顾幸生. 不确定性条件下的生产调度. 华东理工大学学报, 2000, 26(5): 441 – 446.)

[12] YANG Z, LIU C G. A hybrid multi-objective gray wolf optimization algorithm for a fuzzy blocking flow shop scheduling problem. *Advances in Mechanical Engineering*, 2018, 10(3): 1 – 13.

[13] LI K, CHEN J F, FU H, et al. Uniform parallel machine scheduling with fuzzy processing times under resource consumption constraint. *Applied Soft Computing*, 2019, 82.

[14] ZHONG Y G, YANG F, LIU F. Solving multi-objective fuzzy flexible job shop scheduling problem using MABC algorithm. *Journal of Intelligent & Fuzzy Systems*, 2019, 36(2): 1455 – 1473.

[15] WANG Ling, WANG Jingjing, WU Chuge. Advances in green shop scheduling and optimization. *Control and Decision*, 2018, 33(3): 385 – 391.
(王凌, 王晶晶, 吴楚格. 绿色车间调度优化研究进展. 控制与决策, 2018, 33(3): 385 – 391.)

[16] 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. *Journal of Cleaner Production*, 2019, 226: 515 – 525.

[17] CHEN J F, WANG L, PENG Z P. A collaborative optimization algorithm for energy-efficient multi-objective distributed no-idle flowshop scheduling. *Swarm and Evolutionary Computation*, 2019, 50.

[18] RUBINSTEIN R Y. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 1997, 99(1): 89 – 112.

[19] CASERTA M, RICO E Q, URIBE A M. A cross entropy algorithm for the Knapsack problem with setups. *Computers & Operations Research*, 2008 35(1): 241 – 252.

[20] JĘDRZEJOWICZ P, SKAKOVSKI A. A cross-entropy-based population-learning algorithm for discrete-continuous scheduling with continuous resource discretisation. *Neurocomputing*, 2010, 73(4/5/6): 655 – 660.

[21] SANTOSA B, BUDIMAN M A, WIRATNO S E. A cross entropy-genetic algorithm for m-machines no-wait job-shop scheduling problem. *Journal of Intelligent Learning Systems and Applications*, 2011, 3(3): 171 – 180.

[22] WANG G R, LI Q Q, WANG L H. An improved cross entropy algorithm for steelmaking-continuous casting production scheduling with complicated technological routes. *Journal of Central South University*, 2015, 22(8): 2998 – 3007.

[23] LI G C, LIU P, LE C Y, et al. A novel hybrid meta-heuristic algorithm based on the cross-entropy method and firefly algorithm for global optimization. *Entropy*, 2019, 21.

[24] ZITZLER E, LAUMANNS M, THIELE L. SPEA2: Improving the strength pareto evolutionary algorithm. *TIK–Report*, 2001, 103.

[25] ISHIBUCHI H, YOSHIDA T, MURATA T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 204 – 223.

[26] YANG Haijun, LI Jianwu, LI Minqiang. *Evolutionary Algorithms: Schema, Emergence and Hardness*. Beijing: Science Press, 2012.
(杨海军, 李建武, 李敏强. 进化算法的模式、涌现与困难性研究. 北京: 科学出版社, 2012.)

作者简介:

佘明哲 硕士研究生, 目前研究方向为优化调度与智能算法, E-mail: shemingzhe@163.com;

钱 斌 教授, 博士生导师, 目前研究方向为智能调度理论与方法, E-mail: bin.qian@vip.163.com;

胡 蓉 副教授, 硕士生导师, 目前研究方向为优化方法与决策支持系统, E-mail: ronghu@vip.163.com;

吴丽萍 讲师, 硕士生导师, 目前研究方向为智能优化方法, E-mail: 554350754@qq.com;

向凤红 教授, 硕士生导师, 目前研究方向为智能优化方法, E-mail: xiangfh5447@sina.com.