混合离散人工蜂群算法求解含不相关并行机的分布式柔性流水线调度

轩华†，李文婷，李冰

(郑州大学 管理工程学院，郑州市 450001)

摘 要: 研究了每阶段含不相关并行机的分布式柔性流水线调度问题，考虑顺序相关准备时间和工件动态到达时间，以最小化总加权提前/拖期惩罚为目标建立整数规划模型，提出了一种融合离散差分进化算法、变邻域下降算法和局域搜索的混合离散人工蜂群算法以获取近优解。该算法采用基于工厂-工件号的编码以及基于机器最早空闲时间的动态解码机制，通过随机规则和均衡分派策略生成初始工厂-工件序列群，在引领蜂阶段引入离散差分进化算法产生优质工厂-工件序列，在跟随蜂阶段利用变邻域下降算法在被选择序列附近继续搜索以得到邻域序列，在侦察蜂阶段设计基于关键/非关键工厂间插入的局域搜索提高算法搜索能力。通过仿真实验测试不同规模的算例，实验结果表明所提出的混合离散人工蜂群算法表现出较好的求解性能。

关键词: 分布式柔性流水线调度；不相关并行机；混合离散人工蜂群算法；离散差分进化算法

中图分类号: N945 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2021.1438

Hybrid discrete artificial bee colony algorithm for distributed flexible flowline scheduling with unrelated parallel machines

XUAN Hua†, *LI Wenting*, *LI Bing*

(School of Management Engineering，Zhengzhou University，Zhengzhou 450001，China)

Abstract: A distributed flexible flowline scheduling problem with unrelated parallel machines at each stage is studied. Considering sequence-dependent setup times and job dynamic arrival times, an integer programming model is established with the objective of minimizing total weighted earliness and tardiness penalty. A hybrid discrete artificial bee colony algorithm is proposed combined with discrete differential evolution algorithm, variable neighborhood descent algorithm and local search so as to obtain near optimal solutions. In this algorithm, factory-job number based encoding is applied and a dynamic decoding mechanism with the earliest machine idle time is designed. The initial factory-job sequence group is then generated by a random rule and an average assignment strategy. In leading bee phase, the discrete differential evolution algorithm is introduced to yield factory-job sequences with high quality. In following bee phase, a variable neighborhood descent algorithm is used to search around the selected sequences in order to gain neighborhood sequences. Local search based on insertion between critical/non-critical factories is designed to enhance algorithm search ability in scout bee phase. Simulation experiments are performed on different scale problems and testing results demonstrate the proposed hybrid discrete artificial bee colony algorithm has a better resolution performance.

Keywords: Distributed flexible flowline scheduling; unrelated parallel machines; hybrid discrete artificial bee colony algorithm; discrete differential evolution algorithm

0 引 言

在经济全球化、顾客需求多样化和市场竞争不确定性的趋势下，传统制造业正从集中式单工厂生产到分布式多工厂柔性生产模式转变[1]。传统的车间调度主要解决单个工厂的资源配置与优化问题，而分布式车间调度需同时协调多个工厂的生产资源，它已成为生产调度领域的研究热点。在实际生产环境中，并行机的磨损程度不同会影响工件的处理时间，而且同一台机器进行工件转换常需机器清理或更换部件等操作，因此，合理安排工件的机器分配和加工顺序具有重要的现实意义。本文以提前/拖期惩罚的加权和最小化为生产目标，考

收稿日期: 2021-08-16；**修回日期:** 2021-12-30.

基金项目: 国家自然科学基金资助项目 (U1804151, U1604150).

†通讯作者. E-mail: hxuan@zzu.edu.cn.

虑顺序相关准备时间和工件动态到达时间，研究了含不相关并行机的分布式柔性流水线调度问题（Distributed flexible flowline scheduling problem with unrelated parallel machines, DFFSP-UPM）。

近年来，很多学者对FFSP-UPM进行了研究。为解决makespan问题，秦浩翔等[2]提出双层变异迭代贪婪算法；Sun和Gu[3]提出混合分布估计算法。为最小化总加权完工时间，Xuan等[4]考虑恶化工件和顺序相关准备时间，提出了改进离散人工蜂群算法。

在分布式车间调度领域，各分布式工厂包含的机器环境有流水线、柔性流水线、柔性作业车间等。对于分布式阻塞流水线调度，Shao等[5]和Zhang等[6]分别提出了混合增强离散果蝇优化算法和混合离散差分进化算法以解决makespan问题。针对分布式置换流水线调度，Huang等[7]考虑顺序相关准备时间，提出了构造式启发式算法和离散人工蜂群（Discrete artificial bee colony, DABC）算法以最小化makespan；Khare和Agrawal[8]提出了混合离散Harris鹰算法和迭代贪婪算法以最小化总拖期。

FFSP与分布式调度的集成则形成DFFSP。在等同并行机假设下，以最小化makespan为目标，Shao等[9]提出基于分解策略的构造式启发式算法和多邻域迭代贪婪算法。在均匀并行机假设下，Zheng等[10]针对模糊DFFSP提出了结合分布估计算法和迭代贪婪搜索的协同进化算法以同时优化模糊总拖期和鲁棒性。在不相关并行机假设下，Shao等[11]考虑无等待约束，应用多种构造式启发式算法；雷德明等[12]考虑顺序相关准备时间，提出多班教学优化算法以同时最小化makespan和最大延迟时间。Li等[13-14]构建了基于机器位置的数学模型，并提出DABC算法求解了含等同/不相关并行机的DFFSP。

对于分布式柔性作业车间调度，吴锐等[15]设计改进人工蜂群算法以最小化makespan；Li等[16]提出基于帕累托的混合禁忌搜索算法优化makespan、最大负荷、总负荷和提前/拖期惩罚三个问题。

综上所述，关于DFFSP的研究多以makespan为优化目标，机器环境多为等同机，缺少总加权提前/拖期惩罚问题以及不相关并行机环境的综合考量。上述分布式车间调度主要针对所有工件在零时刻可用的静态确定性问题，由于车间地理位置偏差和订单释放的不确定性等，待加工工件进入生产车间之前实际需要消耗一定的时间用于运输等，且已有的关于工件动态到达时间的研究多出现在单机、并行机、流水线、单件车间和开放车间调度[17-18]，故考虑工件动态到达时间的DFFSP-UPM会使所研究问题更贴近实际生产。因此本文提出了融合离散差分进化算法、变邻域下降算法和局域搜索的混合离散人工蜂群（Hybrid discrete artificial bee colony, HDABC）算法以解决含顺序相关准备时间和动态到达时间的DFFSP-UPM的总加权提前/拖期惩罚问题。

1 问题建模

1.1 问题描述

有 F 个位于不同地理位置的相同工厂，每个工厂包含一个由 G 个生产阶段组成的柔性流水线，工厂 f 内阶段 t 由 U_t 台不相关并行机构成，且至少存在一个阶段 $U_t>1$。假设有 N 个相互独立的工件，每个工件 j 在时刻 $r_j\,(r_j>0)$ 到达生产车间的始端，且有一个交货期 d_j。每个工件在任一工厂依次访问 G 个生产阶段，在同一台机器上加工的两个相邻工件之间需要一个准备时间，其大小取决于工件的加工顺序。一台机器一次至多加工一个工件，而一个工件只能在一台机器上加工。根据三域表示法，所研究问题可表述为DFFG(U_1,U_2,$\cdots$,U_G)|P_{jftm}=P_{jtm}|E/T，其中DFFG(U_1,U_2,$\cdots$,U_G)表示每阶段有 U_t 台不相关并行机的分布式柔性流水线，P_{jftm}=P_{jtm} 表示每个工件在任一工厂内阶段 t 机器 m 上处理时间都相同，E/T表示优化目标是总加权提前/拖期惩罚达到最小。

1.2 符号定义

（1）参数

f：工厂标号,$f\in\{1,...,F\}$,F 为工厂数

j：工件标号,$j\in\{1,...,N\}$,N 为工件数

t：生产阶段标号,$t\in\{1,...,G\}$,G 为生产阶段数

m：机器标号,$m\in\{1,...,U_t\}$,U_t 为阶段 t 的不相关并行机数

P_{jtm}：工件 j 在阶段 t 的机器 m 上的处理时间

$S_{j'jtm}$：在阶段 t 的机器 m 上从工件 j' 到工件 j 的准备时间，若在机器 m 上工件 j 紧随工件 j' 之后加工，则 $S_{j'jtm}>0$，否则 $S_{j'jtm}=0$

r_j：工件 j 的动态到达时间

d_j：工件 j 的交货期

L：一个足够大的正数

α_j: 工件 j 的提前惩罚系数

β_j: 工件 j 的延迟惩罚系数

E_j: 工件 j 的提前期

T_j: 工件 j 的拖期

（2）决策变量

B_{jt}：工件 j 在生产阶段 t 的开工时间

C_{jt}：工件 j 在生产阶段 t 的完工时间

CT_j：工件 j 的完工时间

X_{jf}：0-1 二元变量，若工件 j 分配至工厂 f 加工，$X_{jf}=1$，否则，X_{jf} =0

Y_{jftm}：0-1 二元变量，若工件 j 在工厂 f 内阶段 t 的机器 m 上加工，Y_{jftm}=1，否则，Y_{jftm}=0

$Z_{j'jftm}$：0-1 二元变量，若工件 j' 和 j 均分配至工厂 f 内阶段 t 的机器 m 上加工且工件 j' 在工件 j 的紧前面，$Z_{j'jftm}$=1，否则，$Z_{j'jftm}$=0

1.3 模型构建

$$min\,TWET=\sum_{j=1}^{N}(\alpha_jE_j+\beta_jT_j) \tag{1}$$

$$s.t.C_j+E_j-T_j=d_j,\ \forall\, j \tag{2}$$

$$E_j\geq d_j-B_{jG}-\sum_{f=1}^{F}\sum_{m=1}^{U_G}Y_{jfGm}(P_{jGm}+S_{j'jGm}),\quad \forall\, j',j;j'\neq j \tag{3}$$

$$T_j\geq B_{jG}+\sum_{f=1}^{F}\sum_{m=1}^{U_G}Y_{jfGm}(P_{jGm}+S_{j'jGm})-d_j,\quad \forall\, j',j;j'\neq j \tag{4}$$

$$\sum_{f=1}^{F}X_{jf}=1,\ \forall\, j \tag{5}$$

$$X_{jf}=\sum_{m=1}^{U_t}Y_{jftm},\ \forall\, j,f,t \tag{6}$$

$$B_{j1}\geq r_j,\ \forall\, j \tag{7}$$

$$C_{jt}-C_{j,t-1}\geq\sum_{f=1}^{F}\sum_{m=1}^{U_t}Y_{jftm}(P_{jtm}+S_{j'jtm}),\quad \forall\, j',j,t;j'\neq j;t\neq 1 \tag{8}$$

$$Z_{j'jftm}+Z_{jj'ftm}\leq 1,\forall\, j',j,f,t,m;j'\neq j \tag{9}$$

$$C_{jt}\geq C_{j't}+\sum_{f=1}^{F}\sum_{m=1}^{U_t}Y_{jftm}(P_{jtm}+S_{j'jtm})-L\times(3-Y_{j'ftm}-Y_{jftm}-Z_{j'jftm}),\quad \forall\, j',j,t;j'\neq j \tag{10}$$

$$CT_j=C_{jG},\ \forall\, j \tag{11}$$

$$E_j,T_j\geq 0,\ \forall\, j \tag{12}$$

$$B_{jt},C_{jt}\geq 0,\ \forall\, j,t \tag{13}$$

$$X_{jf},Y_{jftm},Z_{j'jftm}\in\{0,1\},\quad \forall\, j',j,f,t,m;j'\neq j \tag{14}$$

式 (1) 为所研究问题的目标，即最小化总加权提前/拖期惩罚；约束 (2) 描述提前期和拖期与交货期之间的对应关系；约束 (3)、(4) 和 (12) 定义提前期和拖期；约束 (5) 说明每个工件只能在一个工厂加工；约束 (6) 确保每个工件只能在一台机器上加工；约束 (7) 说明工件在第一个生产阶段的开工时间需在它到达生产车间之后；约束 (8) 表示同一工厂内一个工件的工序优先级关系；约束 (9)-(10) 确保一台机器在任一时刻至多处理一个工件；约束 (11) 定义工件的完工时间；约束 (13)-(14) 说明决策变量的取值范围。

2 混合离散人工蜂群算法

ABC 算法包含引领蜂、跟随蜂和侦察蜂三个搜索阶段。鉴于混合算法可以平衡搜索的广度和深度，而离散差分进化 (Discrete differential evolution, DDE) 算法具有较强的全局搜索能力，基于问题特征的变邻域下降 (Variable neighborhood descent, VND) 算法能对优质区域进行二次搜索，局域搜索 (Local search, LS) 可替换无法改进的序列以避免无效搜索。因此，对每个阶段离散化改进后提出 HDABC 算法。

2.1 编码和解码

2.1.1 基于工厂-工件号的编码机制

考虑到分布式柔性流水线各个工厂之间的并行与独立性，提出基于工厂-工件号的整数编码机制，将调度解表述为长度为 N 的工厂-工件序列，它由工厂序列 $\tau=\{\tau_f|f=1,2,...,F\}$ 和工件子序列 $\tau_f=[J_{f-1},J_{f-2},...,J_{f-n^f}]$ 组成，其中 τ_f 描述了在工厂 f 内第一个生产阶段加工的工件加工顺序，即它包含了分配至该工厂的工件号以及这些工件的加工顺序信息，且有 $\sum_f n^f=N$。

2.1.2 基于机器最早空闲时间的动态解码机制

在阶段 1 按照已知工件子序列依次安排工件，而后续阶段则根据前一阶段各工件的完工时间和该阶段上机器的最早空闲时间来确定，故提出了一种动态解码机制，具体过程如下：

阶段 1：对于 τ_f 中的第一个工件 J_{f-1}，从可选不相关并行机器集 $\{UM^f_{1-1},...,UM^f_{1-U_1}\}$ 选择处理时间最短的机器 q 作为该工件的加工机器，并从可选机器集内移除，继续按照最短处理时间规则依次为 $J_{f-2},...,J_{f-U_1}$ 分派机器并更新可选机器集直至完成前 U_1 个工件的机器分配，令 $B_{J_{f-j}1}=r_{J_{f-j}}$，记录各机器的最早空闲时间 idl_{1m} 为分配至该机器的工件完工时间。对于工件 $J_{f-j}(j=U_1+1,...,n^f)$，选择具有最小 idl_{1m} 值的机器 q 作为加工该工件的机器，令 $B_{J_{f-j}1}=\min\{r_{J_{f-j}},idl_{1q}+S_{J_{f-(j-1)}J_{f-j}1q}\}$，更新该机器的最早空闲时间 $idl_{1q}=C_{J_{f-j}1}$。

阶段 $t(t>1)$：将 τ_f 包含的工件按照阶段 $t-1$ 的完工时间的升序排列，对于前 U_t 个工件，按照最短处理时间规则依次分配至 U_t 台机器，令

$B_{J_{f-j}t}=C_{J_{f-j},t-1}$，记录 idl_{tm}；对于其余工件 $J_{f-j}(j=U_t+1,\ldots,n^f)$，将其分配至具有最小 idl_{tm} 值的机器 q，令 $B_{J_{f-j}t}=\max\{C_{J_{f-j},t-1},\ idl_{tq}+S_{J_{f-(j-1)}J_{f-j}1q}\}$，更新 $idl_{tq}=C_{J_{f-j}t}$。

给定 τ=[4, 12, 11, 3, 9, 1; 2, 5, 7, 6, 8, 10], 假设 $G=U_t=3$，$S_{j'jtm}$ 仅与工件相关，即可表示为 $S_{j'j}$，r_j 设置为 {3, 2, 2, 1, 2, 3, 2, 3, 2, 3, 2, 1} 工件处理时间和准备时间如表 1-2。以工厂 1 为例说明具体解码过程（工厂 2 同理）：

阶段 1：τ_1=[4, 12, 11, 3, 9, 1]

$B_{41}=r_4=1, C_{41}=B_{41}+\min\{P_{411},P_{412},P_{413}\}=1+3=4, Y_{4113}=1, idl_{13}=C_{41}=4$;

$B_{12,1}=r_{12}=1, C_{12,1}=B_{12,1}+\min\{P_{12,1,1},P_{12,1,2}\}=1+4=5, Y_{12,1,1,1}=1, idl_{11}=C_{12,1}=5$;

$B_{11,1}=r_{11}=2, C_{11,1}=B_{11,1}+\min\{P_{11,1},2\}=2+4=6, Y_{11,1,1,2}=1, idl_{12}=C_{11,1}=6$;

$\min\{idl_{11},idl_{12},idl_{13}\}=4\Rightarrow Y_{3113}=1$

$B_{31}=\min\{r_3,idl_{13}+S_{43}\}=7$

$C_{31}=B_{31}+P_{313}=7+2=9, idl_{13}=C_{31}=9$;

$\min\{idl_{11},idl_{12},idl_{13}\}=5\Rightarrow Y_{9111}=1$

$B_{91}=\min\{r9,idl_{11}+S_{12,9}\}=7$

$C_{91}=B_{91}+P_{911}=7+7=14, idl_{11}=C_{91}=14$;

$\min\{idl_{11},idl_{12},idl_{13}\}=6\Rightarrow Y_{1112}=1$

$B_{11}=\min\{r_1,idl_{12}+S_{11,1}\}=7$

$C_{11}=B_{11}+P_{112}=7+4=11, idl_{12}=C_{11}=11$。

阶段 2：4→12→11→3→1→9

$B_{42}=C_{41}=4, C_{42}=B_{42}+\min\{P_{421},P_{422},P_{423}\}=4+2=6, Y_{4122}=1, idl_{22}=C_{42}=6$;

$B_{12,2}=C_{12,1}=5, C_{12,2}=B_{12,2}+\min\{P_{12,2,1},P_{12,2,3}\}=5+6=11, Y_{12,1,2,1}=1, idl_{21}=C_{12,2}=11$;

$B_{11,2}=C_{11,1}=6, C_{11,2}=B_{11,2}+\min\{P_{11,2,3}\}=6+7=13, Y_{11,1,2,3}=1, idl_{23}=C_{11,2}=13$;

$\min\{idl_{21},idl_{22},idl_{23}\}=6\Rightarrow Y_{3122}=1$

$B_{32}=\max\{C_{31},idl_{22}+S_{43}\}=10$

$C_{32}=B_{32}+P_{322}=10+4=14, idl_{22}=C_{32}=14$;

$\min\{idl_{21},idl_{22},23\}=11\Rightarrow Y_{1121}=1$

$B_{12}=\max\{C_{11},idl_{21}+S_{12,1}\}=15$

$C_{12}=B_{12}+P_{121}=15+4=19, idl_{21}=C_{12}=19$;

$\min\{idl_{21},idl_{22},idl_{23}\}=13\Rightarrow Y_{9123}=1$

$B_{92}=\max\{C_{91},idl_{23}+S_{11,9}\}=15$

$C_{92}=B_{92}+P_{923}=15+5=20, idl_{23}=C_{92}=20$。

阶段 3：4→12→11→3→1→9

$B_{43}=C_{42}=6, C_{43}=B_{43}+\min\{P_{431},P_{432},P_{433}\}=6+\min\{3,4,2\}=8, Y_{4133}=1, idl_{33}=C_{43}=8$;

$B_{12,3}=C_{12,2}=11, C_{12,3}=B_{12,3}+\min\{P_{12,3,1},P_{12,3,2}\}=11+6=17, Y_{12,1,3,2}=1, idl_{32}=C_{12,3}=17$;

$B_{11,3}=C_{11,2}=13, C_{11,3}=B_{11,3}+\min\{P_{11,3,1}\}=13+2=15, Y_{11,1,3,1}=1, idl_{31}=C_{11,3}=15$;

$\min\{idl_{31},idl_{32},idl_{33}\}=8\Rightarrow Y_{3133}=1$

$B_{33}=\max\{C_{32},idl_{33}+S_{43}\}=14, C_{33}=B_{33}+P_{333}=14+8=22, idl_{33}=C_{33}=22$;

$\min\{idl_{31},idl_{32},idl_{33}\}=15\Rightarrow Y_{1131}=1$

$B_{13}=\max\{C_{12},idl_{31}+S_{11,1}\}=19, C_{13}=B_{13}+P_{131}=19+6=25, idl_{31}=C_{13}=25$;

$\min\{idl_{31},idl_{32},idl_{33}\}=17\Rightarrow Y_{9132}=1$

$B_{93}=\max\{C_{92},idl_{32}+S_{12,9}\}=20, C_{93}=B_{93}+P_{932}=20+4=24, idl_{32}=C_{93}=24$。

表 1　处理时间

t	P_{1tm}	P_{2tm}	P_{3tm}	P_{4tm}	P_{5tm}	P_{6tm}	P_{7tm}	P_{8tm}	P_{9tm}	P_{10tm}	P_{11tm}	P_{12tm}
1	2 4 7	5 6 8	9 5 2	9 6 3	3 2 5	5 3 9	8 6 4	2 3 5	7 5 5	9 9 5	8 4 5	4 9 9
2	4 5 6	9 7 5	5 4 3	3 2 5	4 5 8	2 3 4	6 5 7	9 2 6	8 5 5	8 9 5	9 9 7	6 9 8
3	6 9 4	5 3 6	3 5 8	3 4 2	4 3 2	5 6 4	4 6 8	5 4 9	3 4 5	5 6 9	2 9 9	7 6 7

表 2　准备时间

j'	$S_{j'1}$	$S_{j'2}$	$S_{j'3}$	$S_{j'4}$	$S_{j'5}$	$S_{j'6}$	$S_{j'7}$	$S_{j'8}$	$S_{j'9}$	$S_{j'10}$	$S_{j'11}$	$S_{j'12}$
1	0	3	2	3	4	4	5	4	2	5	1	5
2	3	0	1	5	2	5	1	3	3	1	4	1
3	2	2	0	2	1	3	4	1	4	5	1	2
4	3	1	4	0	5	4	1	2	5	3	2	4
5	3	2	2	4	0	3	5	1	3	2	4	3
6	2	4	3	1	4	0	5	2	2	2	2	3
7	4	5	2	5	4	3	0	2	2	1	4	2
8	5	3	4	5	3	3	5	0	3	2	3	5
9	1	5	5	5	2	5	3	5	0	2	5	1
10	4	2	1	5	2	1	3	3	4	0	5	2
11	1	3	2	3	2	4	5	3	2	5	0	3
12	4	3	5	2	2	1	5	3	2	5	2	0

因此，按上述解码过程得到如图 1 的解码结果。

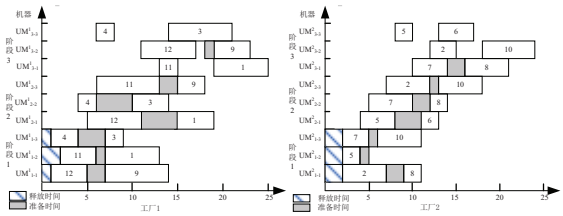

图 1　解码后的调度甘特图

2.2　初始工厂-工件序列群的产生

利用随机规则产生一个包含 N 个工件号的加工序列；然后引入均衡分派策略确定工件至工厂的分配，即取 $\lfloor N/F\rfloor$ 作为每个工厂至少加工的工件数，随机产生 $N-F\cdot\lfloor N/F\rfloor$ 个工厂，这些工厂处理的工件数均为 $\lfloor N/F\rfloor+1$，而其它工厂的工件数均为 $\lfloor N/F\rfloor$，从而最终确定了各工厂加工的工件数 n^f，将加工序列的前 n^1 个工件分配至工厂 1，第 n^1+1 至 n^1+n^2 个位置的工件归入工厂 2，依次类推，第 $n^1+n^2+\ldots+n^{F-1}+1$ 至 $n^1+n^2+\ldots+n^F$ 个位置的工

件归入工厂 F; 最后, 对每个工厂, 将分配至该工厂的工件按照 r_j 升序排列形成最终的工件子序列 τ_f。

2.3 引领蜂搜索阶段

引入 DDE 算法使引领蜂在当前工厂-工件序列群的邻域展开搜索，以获取新工厂-工件序列。

2.3.1 DE/*best*/1 变异

采用“DE/*best*/1”变异方式，其中“DE”表示差分进化算法，“*best*”表示选择最优工厂-工件序列作为变异向量，“1”表示变异过程中差向量的个数为 1。令 $H_a=[h_a(1),...,h_a(k),...,h_a(N)]$ 和 $H_b=[h_b(1),...,h_b(k),...,h_b(N)]$ 为随机选择的两个工厂-工件序列，扩展 Zhang 等[6] 的研究，得到变异算子:

$$V_i^l=H_{best}^{l-1}\oplus\sigma\otimes(H_a^{l-1}-H_b^{l-1})' \tag{15}$$

其中 $a,b,i\in[1,NS]$ (NS 为工厂-工件序列群规模), l 为当前迭代数，σ 为变异影响因子。

变异算子执行如图 2，包括三部分:

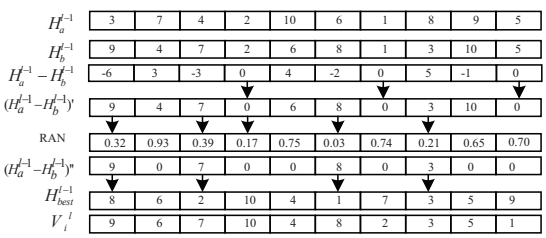

图 2 变异算子

步骤 1: 对 H_a^{l-1} 和 H_b^{l-1} 进行差分算子得到差向量 $(H_a^{l-1}-H_b^{l-1})'$。将 H_a^{l-1} 和 H_b^{l-1} 内各对应元素相减得到向量 $(H_a^{l-1}-H_b^{l-1})$，若其中有非零元素，则用 H_b^{l-1} 中对应位置的工件号取代该非零元素，即

$$\xi_i^l=(H_a^{l-1}-H_b^{l-1})'$$

$$\Leftrightarrow\xi_i^l(k)=\begin{cases}h_b^{l-1}(k), & h_a^{l-1}(k)\neq h_b^{l-1}(k)\\ 0, & \text{其他}\end{cases} \tag{16}$$

其中 $\xi=[\xi(1),\xi(2),...,\xi(N)]$。

步骤 2: 根据 σ 和随机数 $rand$, 对差向量 $(H_a^{l-1}-H_b^{l-1})'$ 做 $\otimes$ 运算得到加权差向量 $(H_a^{l-1}-H_b^{l-1})''$。随机产生一个与工厂-工件序列相同维度的序列 RAN, 序列中元素在 [0, 1] 随机生成。若 $rand<\sigma$，则保留差向量 ξ_i^l 对应位置的元素不变，否则，将差向量 ξ_i^l 对应位置的元素替换为 0，即

$$\gamma_i^l=(H_a^{l-1}-H_b^{l-1})''=\sigma\otimes\xi_i^l$$

$$\Leftrightarrow\gamma_i^l(k)=\begin{cases}\xi_i^l(k), & rand<\sigma\\ 0, & \text{其他}\end{cases} \tag{17}$$

其中 $\gamma=[\gamma(1),\gamma(2),...,\gamma(N)]$。

步骤 3: 将 $(H_a^{l-1}-H_b^{l-1})''$ 与最优工厂-工件序列 H_{best}^{l-1} 执行 $\oplus$ 运算, 得到一个变异工厂-工件序列 V_i^l。首先初始化 $k=1$; 当 $\gamma_i^l(k)=0$ 或 $\gamma_i^l(k)\neq0$ 且 $\gamma_i^l(k)=h_{best}^{l-1}(k)$ 时，保持 $h_{best}^{l-1}(k)$ 不变，当 $\gamma_i^l(k)\neq h_{best}^{l-1}(k)\neq0$ 时，在 h_{best}^{l-1} 中寻找并记录工件号为 $\gamma_i^l(k)$ 所处的位置 k', 交换 h_{best}^{l-1} 中处于位置 k 和 k' 的工件号 $h_{best}^{l-1}(k)$ 和 $h_{best}^{l-1}(k')$, 更新 h_{best}^{l-1}; 最后, 令 $k=k+1$, 若 $k\leq N$, 重复前一过程，否则，输出 $V_i^l=h_{best}^{l-1}$。

2.3.2 单点交叉

对 V_i^l 和前次迭代工厂-工件序列群中的 H_i^{l-1} 应用单点交叉算子。如图 3，交换 H_i^{l-1} 和 V_i^l 内位于交叉点 μ 之前的序列片段，形成子代序列 $W1_i^l=w1_i^l(1),...,w1_i^l(\mu-1),w1_i^l(\mu),...,w1_i^l(N)$ 和 $W2_i^l=w2_i^l(1),...,w2_i^l(\mu-1),w2_i^l(\mu),...,w2_i^l(N)$。记录 $W1_i^l$ 具有重复工件号的位置 μ_1 和 μ_2，将工件号 $w1_i^l(\mu_1)$ 替换成工件号 $w2_i^l(\mu_2)$，重复该过程直至 $W1_i^l$ 没有重复的工件号；对 $W2_i^l$ 也执行上述修正过程。

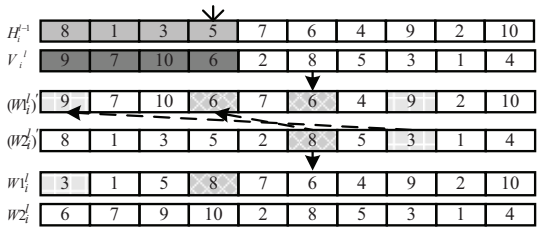

图 3 交叉算子

2.3.3 概率选择

基于模拟退火算法提出以一定接受概率为准则的选择策略，使目标值大的序列也能进入当前迭代序列群成为试验序列 H_i^l。结合 Khare 和 Agrawal[8] 的选择规则设计工厂-工件序列选择公式如下:

$$H_i^l=\begin{cases}W_i^l, & \Delta O<0\text{或}rand'<Exp(-\Delta O<T)\\ H_i^{l-1}, & \text{其他}\end{cases} \tag{18}$$

其中 $\Delta O=TWET(W_i^l)-TWET(H_i^{l-1}),rand'\in(0,1)$，$T$ 为简化后的固定温度值，给定如下:

$$T=\frac{F\times\sum_{j=1}^{N}\sum_{t=1}^{G}\sum_{m=1}^{U_t}P_{jtm}}{10\times N\times F\times\sum_{t=1}^{G}U_t} \tag{19}$$

2.4 引领蜂搜索阶段

跟随蜂采用轮盘赌规则在更新后的工厂-工件序列群中选择工厂-工件序列, 试验序列 H 的适应度值和选择概率分别为

$$fit(H)=1/\sum_{j=1}^{N}(\alpha_jE_j+\beta_jT_j) \tag{20}$$

$$FP(H)=fit(H)/\sum_i fit(i) \tag{21}$$

对所选序列执行 VND 算法，设计三种邻域产生

机制：随机工厂内插入、关键/非关键工厂间交换和关键/非关键工厂内逆序反转。依次执行这三种邻域产生机制，若新邻域序列优于当前序列，则替换当前序列，搜索结束，否则继续执行下一种邻域搜索。

2.4.1 随机工厂内插入

随机选择工厂 f 和工件 $J_{f-\theta}(1 \le \theta \le n^f)$，将 $J_{f-\theta}$ 从 τ_f 中移除，形成片段子序列 $\tau_f^{'}$；将 $J_{f-\theta}$ 逐次插入到 $\tau_f^{'}$ 中的所有可能位置，选择具有最小目标值的子序列作为工厂 f 的最终工件子序列，如图 4。

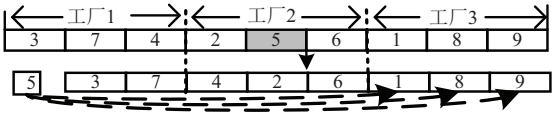

图 4 随机工厂内插入

2.4.2 关键/非关键工厂间交换

设有最大目标值的工厂为关键工厂 $f1$，有最小目标值的工厂 $f3$ 为非关键工厂。从 $f1$ 和 $f3$ 分别随机产生工件 $J_{f1-\theta1}$ 和 $J_{f3-\theta2}$ 并进行交换，如图 5。

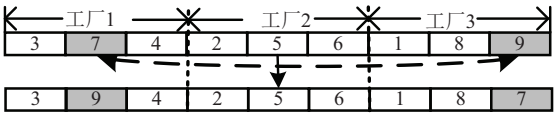

图 5 关键/非关键工厂间交换

2.4.3 关键/非关键工厂内逆序反转

随机从关键工厂 $f1$ 和非关键工厂 $f3$ 各选一个工件 $J_{f1-\sigma1}$ 和 $J_{f3-\sigma2}$，以这两个工件所处位置作为变异点，将位于这两个变异点之间的工件进行逆序反转，如图 6。

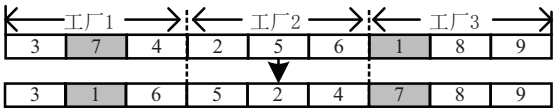

图 6 关键/非关键工厂内逆序反转

2.5 侦察蜂搜索阶段

如果某个工厂-工件序列执行 $limit$ 次迭代后适应度值仍未改进，则引领蜂转变为侦察蜂。侦察蜂通过关键/非关键工厂间插入方式进行 LS 以获取新工厂-工件序列。从关键工厂内随机选择一个工件 J_ρ 并移除，将 J_ρ 依次插入到非关键工厂的工件子序列中的所有可能位置，具有最小目标值的子序列即为非关键工厂的新工件子序列，如图 7。

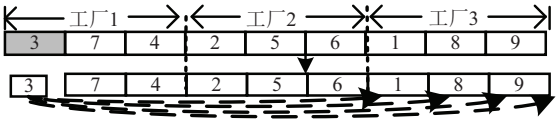

图 7 关键/非关键工厂间插入

2.6 HDABC 算法流程

基于上述描述得到 HDABC 算法的总体流程：

步骤 1：参数初始化，设置工厂-工件序列群规模 NS，最大迭代数 $MaxG$，变异影响因子 σ，最大尝试次数 $limit$；

步骤 2：采用随机规则和均衡分派策略初始化工厂-工件序列群；

步骤 3：计算每个工厂-工件序列的适应度值，记录全局最优序列；

步骤 4：当算法超过 $MaxG$ 时，输出最优工厂-工件序列及其目标值；否则，继续执行步骤 5；

步骤 5：引领蜂执行离散差分进化算法，采用一定接受概率的选择策略产生新工厂-工件序列群；

步骤 6：跟随蜂采用轮盘赌规则选择较优工厂-工件序列，进行变邻域下降算法；

步骤 7：若某个工厂-工件序列在 $limit$ 次迭代后仍未改进，则引领蜂转变为侦察蜂，继续步骤 8；否则，转至步骤 3；

步骤 8：侦察蜂对被放弃列进行局域搜索，产生新工厂-工件序列，转至步骤 3。

3 实验仿真与分析

3.1 算例生成

为了测试 HDABC 算法的性能，随机生成不同规模的算例进行仿真实验。数据产生如下：$F \in \{2,4,5,6\}$，$N \in \{10,20,40,60,90,120\}$，$G \in \{3,4,5\}$，$U_t \in \{3,4,5,6\}$；$P_{jtm} \in U[1,10]$，$S_{j'jtm} \in U[0,5]$，$r_j \in U[1,5]$。定义工件在所有阶段的总平均处理时间 P_j^{avg} 和总平均准备时间 S_j^{avg}：

$$P_j^{avg} = (\sum_{t=1}^{G}\sum_{m=1}^{U_t} P_{jtm}) / \sum_{t=1}^{G} U_t \tag{22}$$

$$S_j^{avg} = \sum_{t=1}^{G} \frac{\sum_{m=1}^{U_t}\sum_{j'=1,j'\le j}^{N} S_{j'jtm}}{U_t \times (N-1)} \tag{23}$$

进而，计算每个工件的交货期[19]：

$$d_j = (P_j^{avg} + S_j^{avg}) \times (1+\lambda\times 3), \lambda \in U[0,1] \tag{24}$$

3.2 实验设计

HDABC 算法与变邻域搜索 (VNS)、传统 ABC 算法 (ABC)、Zhang 等[6] 提出的混合局域搜索的 DDE 算法 (HDDE)、轩华等[20] 设计的自适应遗传算法 (AGA)、Xuan 等[4] 提出的结合 GA、变邻域搜索和贪婪启发式算法的改进 DABC 算法 (IDABC) 进行对比以验证所提算法的求解性能。所有算法均采用 Matlab R2020a 编程，在 Inter Core i5-6200U CPU(2.30GHz) 的微机上运行。为公平比较这六种

算法，通过正交实验设置 σ=0.5，$limit$=5，NS=80；$MaxG$=100；将 ABC 算法运行 100 代且最大 CPU 时间不超过 1000s 所需的 CPU 时间作为其它五种算法的停止时间。

$\{F, N, G, U_t\}$ 的不同组合共产生 56 种问题规模，每种规模随机产生 10 组算例，共测试 560 组算例。定义相对百分比偏差 (%) 为：

$$R(\varepsilon) = \frac{TWET_{\varepsilon} - TWET_{HDABC}}{TWET_{HDABC}} \times 100\% \quad (25)$$

其中 $TWET_{\varepsilon}$ 为当前算法 $\varepsilon(\varepsilon \in \{\text{VNS}, \text{ABC}, \text{HDDE}, \text{AGA}, \text{IDABC}\})$ 求解算例的平均目标值，$TWET_{HDABC}$ 为 HDABC 算法得到的平均目标值。

3.3 HDABC 算法的有效性检验

设置 100 次迭代作为算法停止条件，取 $N = \{40, 60, 90, 120\}$ 且 $F = G = U_t = 4$ 的各规模问题的一组算例，分别运行未引入 DDE 算法的 HDABC 算法 (ABC1)、未结合 VND 算法的 HDABC 算法 (ABC2)、未使用 LS 的 HDABC 算法 (ABC3) 以及 HDABC 算法。如图 8，HDABC 算法的求解效果最好，其它三种算法的收敛性表现均不如 HDABC 算法，因此 DDE、VND 和 LS 这三种算法的引入能提升 HDABC 算法性能。随着问题规模的增大，HDABC 算法与 ABC1 算法的迭代曲线差异较大，与 ABC3 算法的差异较小，因此 DDE 算法对所提算法的影响程度最大，LS 影响最小。

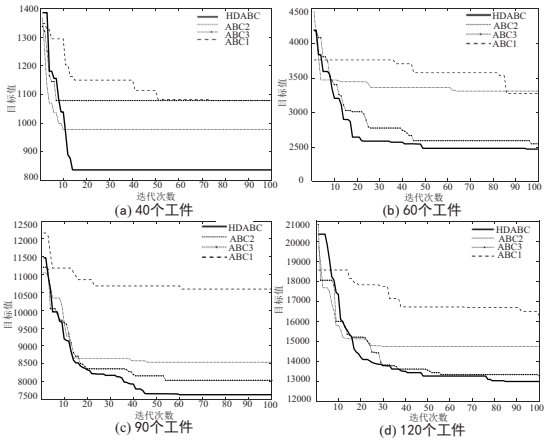

图 8 各规模问题的目标值变化趋势

3.4 测试结果与分析

由前述 HDABC、VNS、ABC、HDDE、AGA 和 IDABC 六种算法求解不同规模算例得到如表 3和表 4所示的测试结果，从表中数据得到以下结论：

(1) 对于中小规模问题, 在平均 CPU 时间 126.4s 内，由 HDABC 算法得到的平均值较 VNS、ABC、HDDE、AGA 和 IDABC 算法分别改进 17.3%、15.2%、11.8%、8.9% 和 4.9%。对于大规模问题的算例，在平均 CPU 时间 641.3 秒内，由 HDABC 算法得到的平均值较 VNS、ABC、HDDE、AGA 和 IDABC 算法分别改进 14.6%、11.9%、10.7%、6.0% 和 2.7%。因此对于所有规模问题，在相同的运行时间内，由 HDABC 算法所获得的解的质量一致优于其它五种算法。

(2) 为了说明实验结果具有统计意义，在 95% 的置信水平下，对实验结果进行了统计学检验。图 9 表明五种对比算法在不同算例测试下所得到的相对百分比偏差结果之间具有显著性差异，且各算法的性能依次按照 HDABC、IDABC、AGA、HDDE、ABC、VNS 降序排列，因此实验结果具有可靠性。

(3) 图 10 显示了参数 N 和 F 对算法性能的影响。随着 N 的增加，问题的复杂度和解空间也相应增大，这导致多数算法的相对百分比偏差呈下降趋势，而且每种对比算法对 N 的变化都比较敏感；随着 F 的增加，F 对算法性能的影响程度不显著，但总体上 HDABC 算法的求解性能最好。

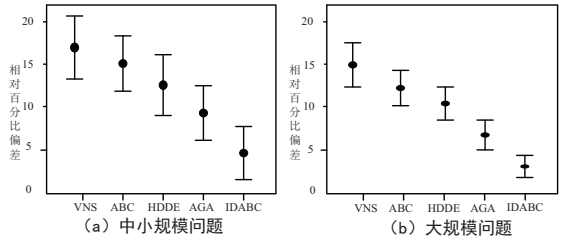

图 9 五种算法求解不同问题的相对百分比偏差区间图

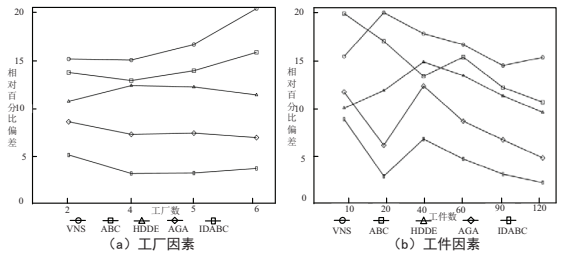

图 10 五种算法和关键因素的交互作用

4 结论

本文研究了考虑不相关并行机和顺序相关准备时间的 DFFSP，目标是最小化总加权提前/拖期惩罚，提出了混合离散人工蜂群算法。结合随机规则和均衡分派策略产生了初始工厂-工件序列群；进而，引领蜂引入 DDE 算法设计了 DE/$best$/1 变异、单点交叉和概率选择算子以得到新序列，跟随蜂采用含随机工厂内插入、关键/非关键工厂间交换和关键/非关键工厂内逆序反转三种邻域产生机制的 VND 算法寻找更优邻域序列，侦查蜂利用 LS 产生新序列替代被放弃序列；最后，对不同规模问题的仿真实验，表明了 HDABC 算法能在可接受的计算时间内得到较好的近优解。未来研究可探讨含能耗约束的多目标 DFFSP-UPM，也可尝试采用其他智能优化算法（如帝国竞争算法等）求解类似问题。

表 3 中小规模问题的测试结果

$F\times N\times G\times U_t$	VNS	ABC	HDDE	AGA	IDABC	HDABC	R1/%	R2/%	R3/%	R4/%	R5/%	CPU/s
2×10×3×3	100.0	109.2	103.8	98.6	98.2	96.8	3.3	13.4	7.2	1.8	1.4	42.2
2×10×3×4	89.0	97.8	81.2	85.0	80.8	79.4	12.0	23.1	2.3	7.1	1.8	49.8
2×10×4×3	85.3	84.2	80.6	83.8	86.4	76.4	11.6	10.2	0.05	9.7	13.1	54.8
2×10×4×4	200.2	196.8	194.2	190.4	176.8	152.2	31.5	29.3	27.5	25.1	16.2	66.1
2×20×3×3	519.2	508.2	488.2	483.0	466.8	455.8	13.9	11.5	7.1	5.9	2.4	77.6
2×20×3×4	280.8	269.2	261.6	255.0	241.4	233.2	20.4	15.4	12.2	9.3	3.5	91.8
2×20×4×3	812.4	794.4	789.4	776.2	756.6	753.2	7.9	5.4	4.8	3.1	0.5	105.1
2×20×4×4	190.8	280.8	295.6	257.6	253.0	248.8	16.9	12.9	18.9	3.5	1.7	122.1
2×40×3×3	3374.6	3249.2	3349.6	3286.4	3216.8	3100.6	8.8	4.8	8.0	5.9	3.7	143.5
2×40×3×4	2072.0	1967.6	2011.2	2037.4	1945.8	1854.2	11.7	6.1	8.5	9.9	4.9	174.2
2×40×4×3	3260.2	3211.2	3172.2	3076.6	3000.2	2900.6	12.4	10.7	9.4	6.1	3.4	190.4
2×40×4×4	2183.8	2032.8	2081.8	1981.6	1902.0	1805.4	20.9	12.6	15.3	9.8	5.4	229.8
4×20×3×3	301.6	287.6	265.0	264.6	249.8	239.0	20.1	20.3	10.9	5.9	4.5	84.1
4×20×3×4	230.0	222.0	204.8	205.6	203.6	203.4	13.1	9.1	0.7	1.1	0.1	93.1
4×20×4×3	290.6	283.6	259.6	252.2	226.8	224.4	29.5	26.4	15.7	12.4	1.1	108.9
4×20×4×4	305.2	298.8	275.6	238.6	243.2	233.0	30.9	28.2	18.3	2.4	4.4	129.5
4×40×3×3	2200.8	2125.6	2157.0	2246.8	2051.0	1793.4	22.7	18.6	20.3	25.3	14.4	149.6
4×40×3×4	1224.6	1167.0	1175.6	1138.2	1064.0	1021.2	19.9	14.3	15.1	11.5	4.2	182.1
4×40×4×3	2554.2	2473.6	2346.2	2229.6	2206.8	2076.6	22.9	19.1	13.0	7.4	6.3	195.4
4×40×4×4	1190.4	1171.4	1256.6	1194.2	1091.4	1025.4	16.1	14.2	22.5	16.5	6.4	237.5
平均	1073.2	1041.5	1042.4	1019.1	978.1	928.7	17.3	15.2	11.8	8.9	4.9	126.4

表 4 大规模问题的测试结果

$F\times N\times G\times U_t$	VNS	ABC	HDDE	AGA	IDABC	HDABC	R1/%	R2/%	R3/%	R4/%	R5/%	CPU/s
4×60×4×3	7281.8	7264.2	7276.7	6990.6	6807.8	6689.0	8.9	8.6	8.8	4.5	1.8	283.1
4×60×4×4	3580.2	3483.8	3577.8	3513.4	3314.8	3227.2	10.9	7.9	10.9	8.9	2.7	358.0
4×60×5×3	6870.0	6807.4	6794.0	6525.8	6363.6	6248.8	9.9	8.9	8.7	4.4	1.8	352.1
4×60×5×4	3783.4	3930.6	3970.0	3647.4	3617.8	3539.0	6.9	11.1	12.2	3.1	2.2	431.5
4×90×4×4	11551.6	11163.2	11931.8	11353.4	10796.4	10784.6	7.1	3.5	10.6	5.3	0.1	518.9
4×90×4×5	7282.2	7385.4	7773.0	7163.2	6965.2	6768.8	7.6	9.1	14.8	5.8	2.9	614.7
4×90×5×4	12972.2	12207.4	12077.6	12334.6	11679.4	11485.2	12.9	6.3	5.2	7.4	1.7	657.0
4×90×5×5	7399.2	7273.0	7255.6	7200.2	6945.2	6848.8	8.0	6.2	5.9	5.1	1.4	776.0
4×120×4×5	14955.6	14794.4	14943.8	14514.4	14061.2	14009.8	6.7	5.6	6.7	3.6	0.3	791.9
4×120×4×6	10436.2	10414.6	10655.6	9718.8	9809.6	9479.4	10.1	9.9	12.4	2.5	3.5	910.4
4×120×5×5	16125.0	15783.8	15706.4	14840.2	14363.2	14321.2	12.5	10.2	9.7	3.6	0.3	1000.0
4×120×5×6	9937.4	9821.8	10306.0	9249.8	9311.4	9302.0	6.8	5.6	10.8	0.5	0.1	1000.0
5×60×4×3	7013.8	6784.4	6863.2	6940.4	6501.6	6293.2	11.4	7.8	9.1	10.3	3.3	288.2
5×60×4×4	3606.0	3756.6	3771.0	3474.4	3309.8	3121.2	15.5	20.4	20.8	11.3	6.0	348.5
5×60×5×3	7647.0	7567.6	7384.8	6760.8	6812.2	6505.4	17.5	16.3	13.5	3.9	4.7	353.7
5×60×5×4	3753.8	3726.6	3770.0	3650.8	3324.2	3142.4	19.4	18.6	20.0	16.2	5.8	435.1
5×90×4×4	11241.6	11284.2	10933.8	10307.4	9993.8	9227.2	21.8	22.3	18.5	11.7	8.3	517.2
5×90×4×5	7029.0	6865.8	6866.2	6520.8	6302.0	6265.8	12.1	9.5	9.6	4.1	0.6	607.9
5×90×5×4	12272.6	12225.0	12114.8	11809.2	11105.4	10614.4	15.6	15.2	14.1	11.3	4.6	644.4
5×90×5×5	7564.2	7550.0	7430.0	6766.8	6713.8	6652.0	13.7	13.5	11.7	1.7	0.9	759.3
5×120×4×5	14678.2	14345.0	14771.8	14053.0	13769.6	13714.4	7.0	4.6	7.7	2.4	0.4	807.1
5×120×4×6	9830.6	9648.2	8749.8	9045.6	8679.2	8637.4	31.8	13.8	1.3	4.7	0.5	919.1
5×120×5×5	15569.6	15138.8	14813.2	14243.2	14103.4	13998.6	11.2	8.1	5.8	1.7	0.7	997.3
5×120×5×6	10980.4	10610.6	10492.8	10331.2	9967.6	9878.0	11.4	7.4	6.2	4.5	0.9	1000.0
6×60×4×3	6694.4	6567.6	6501.8	6315.6	5922.2	5652.2	18.4	16.2	15.0	11.7	4.8	290.4
6×60×4×4	3349.4	3313.2	3161.4	3160.6	2995.8	2779.2	20.5	19.2	13.8	13.7	7.8	353.8
6×60×5×3	6731.4	6542.0	6216.4	5658.8	5891.2	5546.8	21.3	17.9	12.1	2.0	6.2	362.4
6×60×5×4	4300.0	4044.8	3548.6	3500.2	3375.6	3330.6	29.1	21.4	6.5	5.1	1.4	446.7
6×90×4×4	11310.6	11057.0	10652.0	9847.4	9666.2	9380.8	20.6	17.9	13.6	4.9	3.0	514.6
6×90×4×5	7707.2	7315.6	7081.8	6823.0	6688.0	6624.2	16.3	10.4	6.9	2.9	0.9	607.3
6×90×5×4	12416.2	11885.2	12030.0	11209.0	11074.6	10922.6	13.2	8.8	10.1	2.6	1.4	637.6
6×90×5×5	7586.4	7524.0	6957.6	7214.4	6866.0	6602.8	14.9	14.0	5.4	9.3	4.0	748.2
6×120×4×5	16736.2	16257.6	15949.6	14611.8	14880.4	14214.4	17.7	14.3	12.2	2.8	4.7	819.7
6×120×4×6	11688.8	11397.0	11089.4	10718.6	10111.0	9804.8	19.2	16.2	13.1	9.3	3.1	934.2
6×120×5×5	16658.0	16093.0	15903.4	15360.4	14343.8	13852.8	20.2	16.2	14.8	10.9	3.5	1000.0
6×120×5×5	12020.4	10737.8	10614.0	10398.2	10203.2	10067.8	19.4	6.7	5.4	3.3	1.3	1000.0
平均	9460.0	9238.0	9164.8	8771.5	8517.7	8320.4	14.6	11.9	10.7	6.0	2.7	641.3

参考文献 (References)

[1] Jacob L, Rainer L. Production planning and scheduling in multi-factory production networks: a systematic literature review[J]. International Journal of Production Research, 2021, 59(7): 2028-2054.

[2] 秦浩翔, 韩玉艳, 陈庆达, 李俊青, 桑红燕. 求解阻塞混合流水车间调度的双层变异迭代贪婪算法 [J]. 控制与决策, 2021, https://doi.org/10.13195/j.kzyjc.2021.0607.
(Qin H X, Han Y Y, Chen Q D, Li J Q, Sang H Y. A double level mutation iterated greedy algorithm for blocking hybrid flow shop scheduling[J]. Control and Decision, 2021, https://doi.org/10.13195/j.kzyjc.2021.0607.)

[3] Sun Z W, Gu X S. A novel hybrid estimation of distribution algorithm for solving hybrid flowshop scheduling problem with unrelated parallel machine[J]. Journal of Central South University, 2017, 24(8): 1779-1788.

[4] Xuan H, Zhang H X, Li B. An improved discrete artificial bee colony algorithm for flexible flowshop scheduling with step deteriorating jobs and sequence-dependent setup times[J]. Mathematical Problems in Engineering, 2019, 2019: 13 pages.

[5] Shao Z S, Pi D C, Shao W S. Hybrid enhanced discrete fruit fly optimization algorithm for scheduling blocking flow-shop in distributed environment[J]. Expert Systems with Applications, 2020, 145: 113147.

[6] Zhang G H, 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.

[7] Huang J P, Pan Q K, Miao Z H, Gao L. Effective constructive heuristics and discrete bee colony optimization for distributed flowshop with setup times[J]. Engineering Applications of Artificial Intelligence, 2021, 97: 104016.

[8] Khare A, Agrawal S. Effective heuristics and metaheuristics to minimise total tardiness for the distributed permutation flowshop scheduling problem[J]. International Journal of Production Research, 2020: 1-17.

[9] Shao W S, Shao Z S, Pi D C. Modeling and multi-neighborhood iterated greedy algorithm for distributed hybrid flow shop scheduling problem[J]. Knowledge-Based Systems, 2020, 194: 105527.

[10] Zheng J, Wang L, Wang J J. A cooperative coevolution algorithm for multi-objective fuzzy distributed hybrid flow shop[J]. Knowledge-Based Systems, 2020, 194: 105536.

[11] Shao W S, Shao Z S, Pi D C. Effective constructive heuristics for distributed no-wait flexible flow shop scheduling problem[J]. Computers & Operations Research, 2021, https://doi.org/10.1016/j.cor.2021.105482.

[12] 雷德明, 苏斌. 基于多班教学优化的多目标分布式混合流水车间调度 [J]. 控制与决策, 2021, 36(2): 303-313.
(Lei D L, Su B. A multi-class teaching-learning-based optimization for multi-objective distributed hybrid flow shop scheduling[J]. Control and Decision, 2021, 36(2): 303-313.)

[13] Li Y L, Li X Y, Gao L, Zhang B, Pan Q K, Fatih Tasgetiren M, Meng L L. A discrete artificial bee colony algorithm for distributed hybrid flowshop scheduling problem with sequence-dependent setup times[J]. International Journal of Production Research, 2021, 59(13): 3880-3899.

[14] Li Y L, Li X Y, Gao L, Meng L L. An improved artificial bee colony algorithm for distributed heterogeneous hybrid flowshop scheduling problem with sequence-dependent setup times[J]. Computers & Industrial Engineering, 2020, 147: 106638.

[15] 吴锐, 郭顺生, 李益兵, 王磊, 许文祥. 改进人工蜂群算法求解分布式柔性作业车间调度问题 [J]. 控制与决策, 2019, 34(12): 2527-2536.
(Wu R, Guo S S, Li Y B, Wang L, Xu W X. Improved artificial bee colony algorithm for distributed and flexible job-shop scheduling problem[J]. Control and Decision, 2019, 34(12): 2527-2536.)

[16] Li J Q, Duan P Y, Cao J D, Lin X P, Han Y Y. A hybrid Pareto-based tabu search for the distributed flexible job shop scheduling problem with E/T criteria[J]. IEEE Access, 2018, 6: 58883-58897.

[17] 张梓琪, 钱斌, 胡蓉. 混合交叉熵算法求解复杂零等待流水线调度问题 [J]. 控制理论与应用, 2021: 1-15.
(Zhang Z Q, Qian B, Hu R. Hybrid cross-entropy algorithm for solving complex no-wait flow-shop scheduling problem[J]. Control Theory & Applications, 2021: 1-15.)

[18] Al-Shayea A M, Saleh M, Alatefi M, Ghaleb M. Scheduling two identical parallel machines subjected to release times, delivery times and unavailability constraints[J]. Processes, 2020, 8(9): 1025.

[19] Behnamian J, Zandieh M. A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties[J]. Expert Systems with Applications, 2011, 38(12): 14490-14498.

[20] 轩华, 李冰, 罗书敏, 王薛苑. 基于总加权完成时间的可重入混合流水车间调度问题 [J]. 控制与决策, 2018, 33(12): 2218-2226.
(Xuan H, Li B, Luo S M, Wang X Y. Reentrant hybrid flowshop scheduling problem based on total weighted completion time[J]. Control and Decision, 2018, 33(12): 2218-2226.)

作者简介

轩华 (1979−), 女, 教授, 博士, 从事物流优化与控制等研究, E-mail: hxuan@zzu.edu.cn;

李文婷 (1998−), 女, 硕士研究生, 从事生产调度等研究, E-mail: liwentingg1003@163.com;

李冰 (1976−), 男, 教授, 从事运输组织优化等研究, E-mail: lbing@zzu.edu.cn.