文章编号:1671-6833(2022)05-0017-07

具有机器可利用性的双目标置换流水车间调度

轩 华, 李海云, 李 冰

(郑州大学 管理学院, 河南 郑州 450001)

摘 要: 研究了具有机器可利用性的置换流水车间调度问题,引入 CDS 启发式算法和局域搜索构造出改进遗传算法,用于同时最小化总加权完成时间和总加权拖期。应用 CDS 启发式算法产生 40%初始工件加工序列群,其余 60%的初始工件加工序列群则通过随机程序产生,以此来提高初始工件加工序列群的质量。针对交叉和变异之后的工件加工序列,设计基于两两交换、单工件插入和多工件插入 3 种邻域解生成机制的局域搜索,以提高解的搜索空间。将所提出的改进遗传算法与基于遗传算法的 3 种启发式算法进行仿真实验,结果表明:所提算法在平均 77.65 s 内相对于其他算法的目标改进率分别为 5.05%、3.09%、7.33%,这也说明了所提算法在较短的时间内能得到更好的目标值;随着问题规模的增大,改进效果更佳。

关键词: 双目标置换流水车间; 机器可利用性; 释放时间; 改进遗传算法; 邻域解生成机制

中图分类号: TB49 **文献标志码**: A **doi**:10.13705/j.issn.1671-6833.2022.05.003

0 引言

流水车间调度是制造业较为关注的一类研究问题,在工业和许多其他实践中有着广泛的应用[1]。作为其中一类典型问题,置换流水车间调度问题(permutation flow shop scheduling problem, PFSSP)首次由 Johnson[2] 提出,并已被证明是一类典型的 NP-hard 组合优化问题[3]。在现实生产中,由于机器故障或检修等原因导致机器在一定的时间段内是不可用的,这就产生了本文所研究的具有机器可利用性的 PFSSP。

在单目标流水车间调度问题(flow shop scheduling problem, FSSP)的研究中,为最小化最大完工时间(makespan),Mosheiov 等[4] 研究了带单一维修时间窗的 2 台机器流水车间和开放车间调度,对于流水车间,维修发生在第 2 台机器,对于这 2 类问题提出了 3/2 近似算法;为求解 m (m>2)个机器的 FSSP,Miyata 等[5] 提出了构造启发式算法求解带预防性维修和顺序相关设置时间的无等待 FSSP;Aggoune 等[6] 针对带有限机器可利用性的 FSSP,结合两工件的临时几何方法和禁忌搜索算法提出了启发式算法。

在多目标 FSSP 的研究中,周炳海等[7] 考虑了机器的衰退,提出了改进双目标人工蜂群算法用于同时最小化 makespan 和维护成本;Zhang 等[8] 针对带预防性和正确性维修的装配式 PFSSP,提出了新的迭代 Pareto 贪婪算法,用于同时最小化 makespan 和维护成本;Boufellouh 等[9] 针对带预防性维修和全局资源约束的 PFSSP,提出了非支配排序遗传算法和粒子群优化,用于同时最小化 makespan 和总生产成本。

求解 PFSSP 的优化算法多以启发式算法或元启发式算法为主获取问题近优解。为最小化 makespan,Abdel-Basset 等[10] 结合局域搜索策略和鲸鱼优化算法提出了混合鲸鱼优化算法;Zhao 等[3] 提出了一种混合高效作业序列映射方案和变邻域搜索的和声搜索算法;Liu 等[11] 结合 NEH 启发式算法和局域搜索提出了一种混合差分进化算法;Xie 等[12] 提出了一种基于教学-学习的混合优化算法用来分别最小化 makespan 和最大延迟。作为一种智能优化算法,遗传算法已广泛应用于求解车间调度[8] 等领域,而遗传算法与其他算法的结合通常会提高求解性能,因此,在解决双目标 PFSSP 时如何设计有效的遗传算法还要进一步探讨。

就目前查阅的文献而言,关于带机器可利用性的双目标 PFSSP 的研究较为有限,因此,为解决此类 PFSSP,本文引入 CDS 启发式算法、局域搜索、遗传参数随迭代数而变化的自适应调整机

收稿日期:2022-01-06;**修订日期**:2022-03-20

基金项目:国家自然科学基金联合项目(U1804151);河南省科技攻关计划项目(202102310310)

作者简介:轩华(1979—),女,河南睢县人,郑州大学教授,博士,主要从事生产计划与调度、物流优化与控制等研究,E-mail:hxuan@zzu.edu.cn。

制提出了一种改进遗传算法(improved genetic algorithm, IGA),用于解决问题同时最小化总加权完成时间和总加权拖期。

1 问题描述及建模

1.1 问题描述

所研究的PFSSP可描述如下:n个待加工工件须经m台不同的机器进行加工,即每个工件i有m道工序,以相同的加工顺序依次在机器M_1,机器$M_2,\cdots$,机器$M_j,\cdots$,机器M_m上加工e_{ij}个时间段。每台机器j有T个不可用时间窗,且不可用时间窗(G_{jo},H_{jo})事先已知且固定,工件在各机器的加工须在该机器的可利用时间段内进行。每台机器一次至多处理一个工件,而每个工件一次只能在一台机器上加工,工件一旦开始在一台机器上加工,则该工序不允许中断。

1.2 数学模型

$$\min F=\mu F_1+\lambda F_2; \tag{1}$$

$$F_1=\sum_{i=1}^{n}\delta_i C_{im}=\sum_{i=1}^{n}\delta_i(S_{im}+e_{im}); \tag{2}$$

$$F_2=\sum_{i=1}^{n}\delta_i Q_i=\sum_{i=1}^{n}\delta_i\cdot\max(0,S_{im}+e_{im}-d_i); \tag{3}$$

$$C_{\pi_1 j}=a_{\pi_1}+e_{\pi_1 j},\ j=1; \tag{4}$$

$$C_{\pi_1 j}\geqslant C_{\pi_1 j-1}+e_{\pi_1 j},\ j\in(2,3,\cdots,m); \tag{5}$$

$$C_{\pi_i 1}\geqslant C_{\pi_{i-1}1}+e_{\pi_i 1},\ i\in(2,3,\cdots,n); \tag{6}$$

$$C_{\pi_i j}\geqslant\max(C_{\pi_i j-1},C_{\pi_{i-1}j})+e_{\pi_i j},i\in(2,3,\cdots,n),\ j\in(2,3,\cdots,m); \tag{7}$$

$$S_{i1}\geqslant a_i,i\in(1,2,\cdots,n); \tag{8}$$

$$C_{ij}\geqslant S_{ij}+e_{ij},i\in(1,2,\cdots,n),\ j\in(1,2,\cdots,m); \tag{9}$$

$$C_{ij}\leqslant G_{jo}+M(1-X_{ijo}),i\in(1,2,\cdots,n),\ j\in(1,2,\cdots,m),o\in(1,2,\cdots,T); \tag{10}$$

$$C_{ij}\geqslant e_{ij}+H_{jo}(1-X_{ijo}),i\in(1,2,\cdots,n),\ j\in(1,2,\cdots,m),o\in(1,2,\cdots,T); \tag{11}$$

$$\sum_{i=1}^{n}Y_{ik}=1,k\in(1,2,\cdots,n); \tag{12}$$

$$\sum_{k=1}^{n}Y_{ik}=1,i\in(1,2,\cdots,n); \tag{13}$$

$$C_{ij},S_{ij}\geqslant 0,i\in(1,2,\cdots,n),j\in(1,2,\cdots,m); \tag{14}$$

$$X_{ijo},Y_{ik}\in(0,1),i,k\in(1,2,\cdots,n),\ j\in(1,2,\cdots,m),o\in(1,2,\cdots,T)。 \tag{15}$$

其中,式(1)为所研究问题的目标,即最小化总加权完成时间和总加权拖期的线性组合,两部分分别由式(2)和式(3)计算得到,其中μ、λ分别为与完成时间和拖期相关的系数,且$\mu+\lambda=1$;δ_i为工件i的目标权重;S_{im}、C_{im}、e_{im}分别为工件i在机器m上的开始时间、完工时间和加工时间;d_i为工件i的交货期。对于序列π中的第一个工件π_1,约束(4)定义了在第一台机器上的完成时间,其中a_{π_1}为工件π_1的释放时间。约束(5)表示在相邻工序间的加工优先级关系。对于工件加工序列的其他工件$\pi_i(i=2,3,\cdots,n)$,约束(6)表示在第一台机器的加工紧随工件π_{i-1}之后进行。约束(7)描述了在后续机器上工件π_i的完成时间由它在前道工序的完成时间以及其紧前工件π_{i-1}在该机器的完成时间来决定。约束(8)确保了工件在第一台机器的开始时间须在其释放时间之后。约束(9)说明了同一个工件的开始和完成时间的关系。约束(10)和(11)确保了从工件开始加工到结束加工的整个时间段在机器的可用时间段内,其中M为一个无穷大的数,X_{ijo}为二元变量,当工件i在机器j的第o个不可用时间窗之前加工时,$X_{ijo}=1$,否则,$X_{ijo}=0$。约束(12)表示序列的任一位置只能安排一个工件,其中Y_{ik}为二元变量,当工件i分配到加工序列的位置k时,$Y_{ik}=1$,否则,$Y_{ik}=0$。约束(13)表示每个工件只能分派到序列内的一个位置。约束(14)和(15)定义了变量取值范围。

上述模型虽与文献[13]类似,但也有不同:文献[13]针对的是两机FSSP,机器不可用时间段仅发生在第一台机器,目标是最小化最大完工时间,而本文研究的是多机问题,每台机器均须考虑机器可利用性约束,目标的最小化总加权完成时间和总加权拖期。

2 改进遗传算法

2.1 PFSSP的编码及解码

采用工件编号的整数编码对工件加工序列进行编码。基于该加工序列调度工件用来解码,从机器M_1至机器M_m,安排各机器上的工件加工时,检查每个工件从S_{ij}开始的e_{ij}个时间段是否出现在(G_{jo},H_{jo})内,如没有,则确定该S_{ij}值;否则,需将S_{ij}延后,直至$(S_{ij},S_{ij}+e_{ij})$在机器可利用时间段内。

P个工件加工序列即构成初始工件加工序列群。利用CDS启发式算法产生40%的初始工件加工序列群,其余的则通过随机程序产生。CDS启发式算法实现过程如下。

Step 1 将m台机器进行临近两两分组,构成$m-1$个的虚拟两机组合:[1,m]、[(1,2),($m-1$,m)]、[(1,2,3),($m-2$,$m-1$,m)],…,[(1,2,

$\cdots,l),(m+1-l,\cdots,m)]$。

Step 2　对每一个虚拟两机组合,分别计算工件 i 在虚拟两机的总加工时间作为其在相应虚拟机器的模拟加工时间,再利用文献[2]算法获得最优加工序列,从而得到 $m-1$ 个最优加工序列。

Step 3　若 $(m-1)<40\%P$,则将上述过程产生的加工序列重复,以填补 $40\%P-(m-1)$ 个加工序列。

2.2　适应度计算

由于本文优化的目标是最小化总加权完成时间和总加权拖期的加权和,故第 π 个工件加工序列的适应度函数表示为

$$f_\pi=\frac{1}{\mu\sum_{i=1}^{n}\delta_i C_{im}+\lambda\sum_{i=1}^{n}\delta_i Q_i},\pi=1,2,\cdots,P。\quad(16)$$

采用轮盘赌规则选择适应度值较高的工件加工序列执行后续的交叉变异算子。

2.3　交叉和变异

设计自适应交叉概率 P_c 和自适应变异概率 P_m 以改善算法的搜索能力和收敛效果。定义当前工件加工序列群的平均适应度值 f_v:

$$f_v=\sum_{\pi=1}^{P}\frac{f_\pi}{P}。\quad(17)$$

令 MG 为最大迭代次数,g 为当前迭代次数,f_x 为最大适应度值,P_{cmax} 和 P_{cmin} 为交叉概率的最大值和最小值,P_{mmax} 和 P_{mmin} 为变异概率的最大值和最小值。

(1)当 $f_\pi\geqslant f_v$ 时,即当前加工序列的适应度值较大时,P_c 和 P_m 的取值随当前迭代次数的变化而变化。

$$P_c=P_{cmax}-(P_{cmax}-P_{cmin})\left(\frac{g}{MG}+\frac{f_\pi}{f_x-f_v}\right);\quad(18)$$

$$P_m=P_{mmin}+(P_{mmax}-P_{mmin})\left(\frac{g}{MG}+\frac{f_\pi}{f_x-f_v}\right)。\quad(19)$$

(2)当 $f_\pi<f_v$ 时,$P_c=P_{cmax}$,$P_m=P_{mmin}$。即当前加工序列的适应度值较小时,P_c 取交叉概率的最大值 P_{cmax},P_m 取变异概率的最小值 P_{mmin}。

对轮盘赌规则选择操作之后产生的序列群选择连续的两个父代工件加工序列 A 和 B,从(0, 1)随机产生一个数 γ,当 $\gamma<P_c$ 时,采用如图 1 所示的单点交叉方式得到子代工件加工序列。随机产生一个切点,将 A 和 B 中位于该切点后的工件片段进行交换。

对交叉操作后得到的新序列群内的每个工件加工序列,从(0,1)随机生成一个数 u,若 $u<P_m$,采用如图 2 所示的翻转逆序变异方式以获取新工件加工序列。随机生成 2 个切点,将位于这 2 个切点间的工件片段进行翻转。

图 1　单点交叉

Figure 1　Single-point crossover

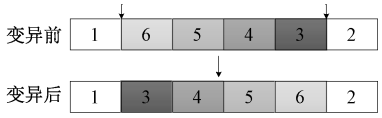


图 2　翻转逆序变异

Figure 2　Reverse order mutation

2.4　局域搜索

对执行上述操作后目标值仍未改进的工件加工序列 Z 依次执行基于如下邻域搜索机制的局域搜索,当变换之后的邻域解优于当前解时,记录最佳解,搜索结束,否则保留当前解。

(1)两两交换。随机选择工件加工序列的 2 个工件编号,将其进行交换,如图 3 所示。

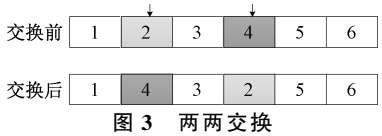


图 3　两两交换

Figure 3　Pair-wise exchange

(2)单工件插入。随机产生工件加工序列的 2 个工件位,将工件位靠前的工件编号插入到工件位靠后的工件编号的前面,如图 4 所示。

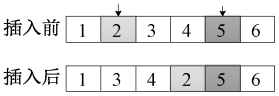


图 4　单工件插入

Figure 4　Single-job insertion

(3)多工件插入。随机选择工件加工序列的 1 个工件片段,从除去该工件片段外的部分工件加工序列里随机选择 1 个工件位,将所除去的工件片段插入至该位置,如图 5 所示。

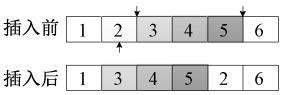


图 5　多工件插入

Figure 5　Multiple-job insertion

2.5　改进遗传算法的计算流程

改进遗传算法具体描述如下。

Step 1　算法参数初始化。工件加工序列群规模 P,最大迭代次数 MG,交叉概率 P_c,变异概率 P_m。

Step 2　工件加工序列群初始化。由 CDS 启发式算法和随机程序的两阶段过程构造初始加工

序列群。

Step 3 计算每个工件加工序列的适应度值,应用轮盘赌规则选择出优秀的工件加工序列。

Step 4 基于自适应交叉概率 P_c 进行单点交叉。

Step 5 基于自适应变异概率 P_m 进行翻转逆序变异。

Step 6 判断经交叉和变异后的工件加工序列的适应度值较原工件加工序列是否有所改善,对未改善的工件加工序列依次基于两两交换、单工件插入和多工件插入产生邻域解,若发现邻域解优于当前解或3种邻域搜索机制执行完毕,结束局域搜索。

Step 7 判断当前迭代数是否超过 MG,如超过,输出最佳目标值;反之,更新当前迭代数,转至Step 3。

3 仿真实验

为了公平起见,将所提出的结合CDS启发式算法和局域搜索的改进遗传算法(IGA)与求解PF-SSP的混合遗传算法(HGA)[14]、不考虑自适应遗传参数设置的改进遗传算法(N-IGA)、结合NEH启发式算法的自适应遗传算法(AGA&NEH)[15]进行比较。所有实验均利用MATLAB 2020a编译,在内存为4.00 GB,处理器为AMD A10-9600P,主频为2.4 GHz的计算机上运行。经实验测试,将N-IGA的交叉概率 P_c 和变异概率 P_m 分别设为0.65和0.02,所有算法的工件加工序列群规模均为100,以HGA运行1 500代且最大运行时间不超过120 s作为其他3种算法的停止条件。

用仿真实验对随机数据测试不同 P_c 和 P_m 取值,将得到的最佳解所对应的 P_c 和 P_m 值作为其最终取值:$(P_{cmin},P_{cmax})=(0.4,0.8)$,$(P_{mmin},P_{mmax})=(0.05,0.1)$。

3.1 参数设置

借鉴文献[16],μ 取2个不同值,分别为 $\mu=0.3$,$\mu=0.7$。令 R 表示交货期的松弛度,取3个不同值,分别为 $R=0.5$,$R=1.0$,$R=1.5$[17]。$n=(50,100,150)$,$m=(5,10)$,a_i、δ_i 和 e_{ij} 分别在(1,5)、(1,4)、(1,20)随机产生。简单起见,假定每台机器有一个不可用时间段,即 $o=1$,但所提出的算法也可以求解机器有多个不可用时间段的情况。

另外,$d_i=R\cdot\sum_j e_{ij}$,机器 M_j 的不可用时间段 GH_j 的长度[13]设置为所有工件在该机器加工的平均时间,即 $GH_j=\sum_i e_{ij}/n$。机器 M_j 的不可用时间段的开始时间和结束时间分别为 $G_{jo}=\alpha\cdot\sum_i e_{ij}-0.5\sum_i e_{ij}/n+H_{j,o-1}$ 和 $H_{jo}=G_{jo}+GH_{jo}=\alpha\cdot\sum_i e_{ij}+0.5\sum_i e_{ij}/n$,其中,$\alpha$ 为满足(0.2,0.8)的均匀分布,由于每次随机产生的 α 不一定相同,故每台机器的不可用时间段也不尽相同。

3.2 IGA的效果分析

为说明CDS启发式算法和基于3种邻域搜索机制的局域搜索对IGA的影响,取 $n=(50,100,150)$,$m=10$,$\mu=0.3$,$R=(0.5,1.0,1.5)$,将IGA、不采用局域搜索的结合CDS启发式和自适应遗传参数的遗传算法(CDS-AGA)和利用随机程序产生初始加工序列群的结合局域搜索和自适应遗传参数的遗传算法(LS-AGA)进行对比。上述参数共产生了9个问题组合,取每个组合10次运行结果的平均值作为该组合的测试结果,因此,本实验共测试了90组实验数据。以CDS-AGA算法运行1 500代且最大运行时间不超过120 s作为其他2种算法的停止条件。测试结果如表1所示,其中加粗显示的为最佳目标值。

表1 IGA、LS-AGA、CDS-AGA算法的测试结果

Table 1 Test results of IGA, LS-AGA, CDS-AGA

(n,m,μ,R)	min F			运行时间/s
	CDS-AGA	LS-AGA	IGA	
(50,10,0.3,0.5)	55 757.05	44 718.95	**44 368.75**	48.75
(50,10,0.3,1.0)	51 643.60	39 988.80	**39 531.70**	49.81
(50,10,0.3,1.5)	47 192.65	35 304.75	**34 474.65**	49.06
(100,10,0.3,0.5)	172 209.40	134 528.30	**132 908.60**	84.36
(100,10,0.3,1.0)	163 902.30	125 675.40	**123 864.50**	84.78
(100,10,0.3,1.5)	153 450.10	115 974.70	**115 228.60**	84.98
(150,10,0.3,0.5)	390 412.45	305 188.25	**303 987.65**	120.00
(150,10,0.3,1.0)	374 156.30	290 400.20	**289 922.70**	120.00
(150,10,0.3,1.5)	352 065.55	277 605.65	**274 687.85**	120.00
平均	195 643.27	152 153.89	**150 997.22**	84.64

由表 1 可看出,IGA 的目标值均优于 CDS-AGA 和 LS-AGA,说明在相同运行时间下,嵌入的 3 种邻域搜索机制可产生更佳邻域解,应用 CDS 启发式算法能够获得更好的初始加工序列群。

3.3　测试结果与分析

(n,m,μ,R)的不同取值共产生了 36 个问题组合,每个组合随机运行 10 次,取其平均值作为该问题组合的测试结果,因此,本实验共测试了 360 个实例。定义 γ_1、γ_2、γ_3 为目标改进率分别为

$$\gamma_1 = (F_{\mathrm{HGA}} - F_{\mathrm{IGA}})/F_{\mathrm{IGA}} \times 100\%;$$

$$\gamma_2 = (F_{\mathrm{N\text{-}IGA}} - F_{\mathrm{IGA}})/F_{\mathrm{IGA}} \times 100\%;$$

$$\gamma_3 = (F_{\mathrm{AGA\&NEH}} - F_{\mathrm{IGA}})/F_{\mathrm{IGA}} \times 100\%。$$

式中:F 表示目标值。

(1)对不同规模问题的测试结果见表 2。由表 2 可知,在平均计算时间 77.65 s 内,HGA、N-IGA、AGA&NEH、IGA 的目标值分别为 148 349.00、145 529.86、151 117.96、140 666.61,IGA 相较于 HGA 改进了 5.05%,较 N-IGA 改进了 3.09%,较 AGA&NEH 改进了 7.33%。

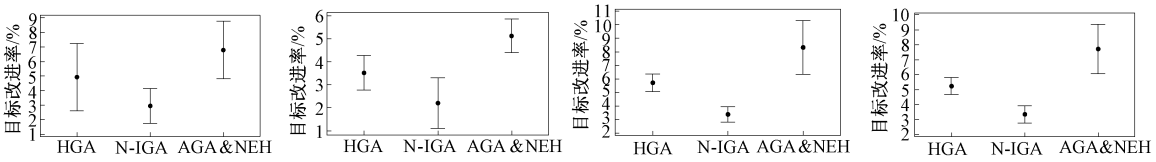
(2)为了对实验结果进行统计意义下的分析和比较,取 $n=(50,100,150)$ 且 $m=(5,10)$ 的不同规模问题,对 IGA 相较于其他 3 种算法的目标改进率进行统计学检验,如图 6 和图 7 所示,IGA 相较于其他 3 种算法均有明显改进。

取 $n=(50,100,150)$,$m=10$,$\mu=0.3$,$R=0.5$ 的 4 个算例在最大迭代次数 1 500 的限制条件下运行上述 4 种算法,得到如图 8 的趋势图,说明在相同迭代数内,IGA 得到的目标值更优。

表 2　不同规模问题的测试结果

Table 2　Test results of different scale problems

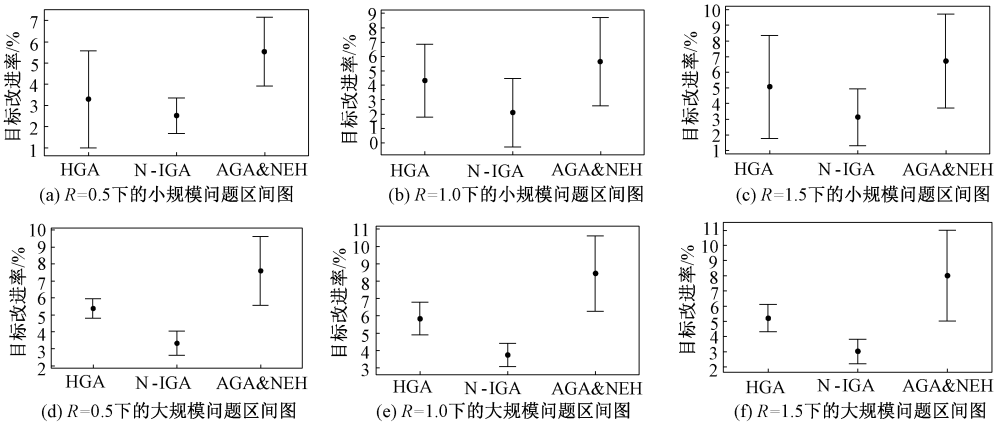
(n,m,μ,R)	min F				γ_1/%	γ_2/%	γ_3/%	运行时间/s
	HGA	N-IGA	AGA&NEH	IGA				
(50,5,0.3,0.5)	30 528.40	29 808.90	30 964.30	29 060.40	5.05	2.58	6.55	40.98
(50,5,0.3,1.0)	28 344.70	27 691.90	28 918.00	26 682.20	6.23	3.78	8.38	39.21
(50,5,0.3,1.5)	26 325.70	25 523.70	26 648.60	24 359.00	8.07	4.78	9.40	41.15
(50,5,0.7,0.5)	31 498.00	31 375.90	32 044.40	30 399.10	3.61	3.21	5.41	39.84
(50,5,0.7,1.0)	30 823.50	30 017.40	31 228.80	29 937.20	2.96	0.27	4.31	39.45
(50,5,0.7,1.5)	29 835.70	29 121.50	29 978.50	28 488.10	4.73	2.22	5.23	39.43
(50,10,0.3,0.5)	45 147.45	45 444.95	46 288.55	44 433.75	1.61	2.28	4.17	54.13
(50,10,0.3,1.0)	41 280.20	39 937.10	41 499.30	39 306.00	5.02	1.61	5.58	53.82
(50,10,0.3,1.5)	35 704.15	35 381.05	36 752.45	34 484.35	3.54	2.60	6.58	53.11
(50,10,0.7,0.5)	48 325.95	47 905.45	49 785.15	46 983.25	2.86	1.96	5.96	53.49
(50,10,0.7,1.0)	46 368.40	46 199.20	46 919.10	45 001.20	3.04	2.66	4.26	53.26
(50,10,0.7,1.5)	44 563.75	44 123.75	45 287.45	42 894.85	3.89	2.86	5.58	53.92
(100,5,0.3,0.5)	124 054.40	120 044.60	123 654.00	117 013.90	6.02	2.59	5.67	65.88
(100,5,0.3,1.0)	119 211.50	115 569.00	119 553.50	111 582.70	6.84	3.57	7.14	65.58
(100,5,0.3,1.5)	113 449.50	110 169.50	115 225.40	108 401.30	4.66	1.63	6.30	65.79
(100,5,0.7,0.5)	125 591.60	124 206.50	126 790.00	119 134.80	5.42	4.26	6.43	68.36
(100,5,0.7,1.0)	124 229.60	121 077.00	124 627.80	116 799.50	6.36	3.66	6.70	67.24
(100,5,0.7,1.5)	121 215.20	119 486.70	122 996.00	117 383.60	3.26	1.79	4.78	70.31
(100,10,0.3,0.5)	140 637.30	137 343.80	148 010.90	133 940.90	5.00	2.54	10.50	95.42
(100,10,0.3,1.0)	130 543.50	128 367.00	140 383.60	124 689.40	4.69	2.95	12.59	93.04
(100,10,0.3,1.5)	121 234.00	119 281.40	134 044.30	115 712.70	4.77	3.08	15.84	92.76
(100,10,0.7,0.5)	144 027.30	141 073.50	154 355.30	137 702.00	4.59	2.45	12.09	94.12
(100,10,0.7,1.0)	140 657.50	138 387.50	152 276.80	135 028.30	4.17	2.49	12.77	92.68
(100,10,0.7,1.5)	138 283.70	134 388.00	145 177.30	131 353.40	5.28	2.31	10.52	93.30
(150,5,0.3,0.5)	248 356.90	244 192.00	251 718.50	233 385.10	6.42	4.63	7.86	93.65
(150,5,0.3,1.0)	241 661.30	237 760.70	245 449.50	229 569.00	5.27	3.57	6.92	90.83
(150,5,0.3,1.5)	234 556.30	229 907.30	235 320.60	221 617.90	5.84	3.74	6.18	90.98
(150,5,0.7,0.5)	250 414.10	247 827.10	254 947.10	238 419.00	5.03	3.95	6.93	91.40
(150,5,0.7,1.0)	249 025.80	244 863.80	250 603.40	234 153.50	6.35	4.57	7.03	91.82
(150,5,0.7,1.5)	246 101.70	242 415.30	248 203.40	232 870.50	5.68	4.10	6.58	90.25
(150,10,0.3,0.5)	318 078.15	314 112.35	321 650.15	304 158.05	4.58	3.27	5.75	120.04
(150,10,0.3,1.0)	309 432.00	302 126.40	309 499.60	287 870.90	7.49	4.95	7.51	120.05
(150,10,0.3,1.5)	293 240.35	284 913.25	294 884.55	274 146.15	6.96	3.93	7.56	120.05
(150,10,0.7,0.5)	330 125.85	320 535.35	329 131.35	311 699.15	5.91	2.83	5.59	120.06
(150,10,0.7,1.0)	322 645.50	318 462.80	326 828.10	305 769.50	5.52	4.15	6.89	120.05
(150,10,0.7,1.5)	315 044.95	310 033.35	318 600.85	299 567.35	5.17	3.49	6.35	120.04
平均	148 349.00	145 529.86	151 117.96	140 666.61	5.05	3.09	7.33	77.65



(a) μ=0.3下的小规模问题区间图 (b) μ=0.7下的小规模问题区间图 (c) μ=0.3下的大规模问题区间图 (d) μ=0.7下的大规模问题区间图

图6 不同μ取值下的目标改进率区间图(95%的置信区间)

Figure 6 Interval diagram of target improvement rate for different μ value (95% confidence interval)

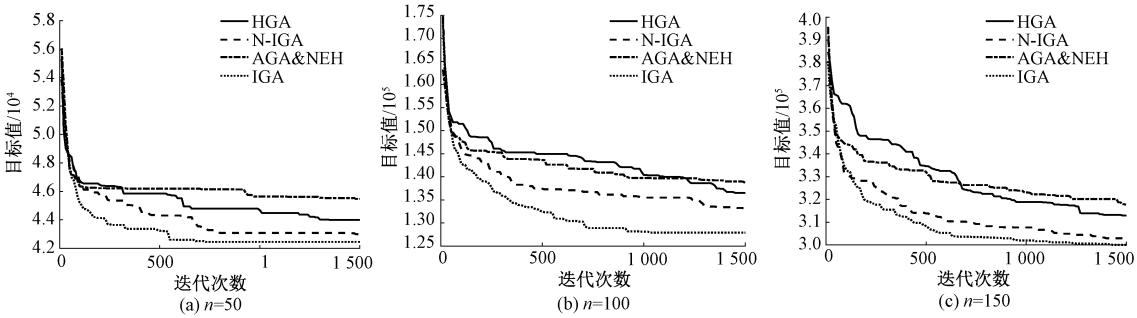


(a) R=0.5下的小规模问题区间图 (b) R=1.0下的小规模问题区间图 (c) R=1.5下的小规模问题区间图

(d) R=0.5下的大规模问题区间图 (e) R=1.0下的大规模问题区间图 (f) R=1.5下的大规模问题区间图

图7 不同R取值下的目标改进率区间图(95%的置信区间)

Figure 7 Interval diagram of target improvement rate for different R value (95% confidence interval)



(a) n=50 (b) n=100 (c) n=150

图8 随迭代次数变化的目标值变化趋势

Figure 8 Trend of the target value changing with the number of iterations

4 结束语

本文研究了具有机器可利用性和工件释放时间的PFSSP。为尽可能最小化总加权完成时间和总加权拖期的加权和,提出了IGA进行求解不同规模的PFSSP,通过仿真实验说明了IGA能够在可接受的计算时间内得到较好的近优解。未来研究可将所提算法推广以求解FSSP或研究其他启发式算法(人工蜂群算法等)求解具有机器可利用性的PFSSP。

参考文献:

[1] YENISEY M M, YAGMAHAN B. Multi-objective permutation flow shop scheduling problem: literature review, classification and current trends[J]. Omega, 2014, 45: 119-135.

[2] JOHNSON S M. Optimal two-and three-stage production schedules with setup times included[J]. Naval research logistics quarterly, 1954, 1(1): 61-68.

[3] ZHAO F Q, LIU Y, ZHANG Y, et al. A hybrid harmony search algorithm with efficient job sequence scheme and variable neighborhood search for the permutation flow shop scheduling problems[J]. Engineering applications of artificial intelligence, 2017, 65: 178-199.

[4] MOSHEIOV G, SARIG A, STRUSEVICH V A, et al. Two-machine flow shop and open shop scheduling problems with a single maintenance window[J]. European journal of operational research, 2018, 271(2): 388-400.

[5] MIYATA H H, NAGANO M S, GUPTA J N D. Integrating preventive maintenance activities to the no-wait flow shop scheduling problem with dependent-sequence setup times and makespan minimization[J]. Computers & industrial engineering, 2019, 135: 79-104.

[6] AGGOUNE R, PORTMANN M C. Flow shop scheduling problem with limited machine availability: a heuristic approach[J]. International journal of production economics, 2006, 99(1/2): 4-15.

[7] 周炳海, 刘子龙. 考虑衰退的流水车间生产与预防性维护集成调度方法[J]. 计算机集成制造系统, 2016, 22(5): 1272-1278.
ZHOU B H, LIU Z L. Integrated scheduling method of flow shop production and preventive maintenance considering decline[J]. Computer integrated manufacturing systems, 2016, 22(5): 1272-1278.

[8] ZHANG Z K, TANG Q H, CHICA M. Maintenance costs and makespan minimization for assembly permutation flow shop scheduling by considering preventive and corrective maintenance[J]. Journal of manufacturing systems, 2021, 59: 549-564.

[9] BOUFELLOUH R, BELKAID F. Bi-objective optimization algorithms for joint production and maintenance scheduling under a global resource constraint: application to the permutation flow shop problem[J]. Computers & operations research, 2020, 122: 104943.

[10] ABDEL-BASSET M, GUNASEKARAN M, EL-SHAHAT D, et al. A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem[J]. Future generation computer systems, 2018, 85: 129-145.

[11] LIU Y, YIN M H, GU W X. An effective differential evolution algorithm for permutation flow shop scheduling problem[J]. Applied mathematics and computation, 2014, 248: 143-159.

[12] XIE Z P, ZHANG C Y, SHAO X Y, et al. An effective hybrid teaching-learning-based optimization algorithm for permutation flow shop scheduling problem[J]. Advances in engineering software, 2014, 77: 35-47.

[13] BENTTALEB M, HNAIEN F, YALAOUI F. Two-machine job shop problem under availability constraints on one machine: makespan minimization[J]. Computers & industrial engineering, 2018, 117: 138-151.

[14] 轩华, 秦莹莹, 王薛苑,等. 带恶化工件的 PFS 调度的混合遗传算法[J]. 工业工程与管理, 2017, 22(3): 1-6,15.
XUAN H, QIN Y Y, WANG X Y, et al. Hybrid genetic algorithm for PFS scheduling with deteriorating artifacts[J]. Industrial engineering and management, 2017, 22(3): 1-6,15.

[15] 轩华, 罗书敏, 王薛苑. 可重入混合流水车间调度的改进遗传算法[J]. 现代制造工程, 2019(2): 18-23,35.
XUAN H, LUO S M, WANG X Y. Improved genetic algorithm for reentrant hybrid flow shop scheduling [J]. Modern manufacturing engineering, 2019(2): 18-23,35.

[16] SHAHVARI O, LOGENDRAN R. Hybrid flow shop batching and scheduling with a bi-criteria objective [J]. International journal of production economics, 2016, 179(3): 239-258.

[17] YAZDANI M, ALETI A, KHALILI S M, et al. Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem[J]. Computers & industrial engineering, 2017, 107: 12-24.

Bi-objective Permutation Flow Shop Scheduling with Machine Availability

XUAN Hua, LI Haiyun, LI Bing

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

Abstract: A permutation flow shop scheduling problem with machine availability was studied. An improved genetic algorithm was proposed by introducing CDS heuristic algorithm and local search so that the total weighted completion time and total weighted tardiness were minimized. To improve the quality of the initial job processing sequence group, the CDS heuristic algorithm is applied to generate 40% of the group and the remaining 60% of the initial job processing sequence group was yielded by random procedure. For the job processing sequence after crossover and mutation, three generation schemes of neighborhood solutions based on pair-wise exchange, single-job insertion and multiple-job insertion were designed to carry out local search in order to extend the search space. The proposed improved genetic algorithm was tested with three genetic algorithm based heuristic algorithms. The results showed that the target improvement rate of the proposed algorithm was 5.05%, 3.09% and 7.33% in the average 77.65 s, compares with other algorithms. It also showed that the proposed algorithm could obtain better target values in a shorter time. With the increase of the problem scale, the improvement effect was better.

Keywords: bi-objective permutation flow shop; machine availability; release time; improved genetic algorithm; generationschemes of neighborhood solutions