基于等级邻域策略的偏序柔性车间优化调度*

李长云① 林 多② 何频捷① 谷鹏飞②
（①湖南工业大学计算机学院，湖南 株洲 412007；②湖南工业大学轨道交通学院，湖南 株洲 412007）

摘 要：针对服装生产车间中工件工艺流程具有复杂偏序关系的调度问题，分析了该类问题的特点和难点，采用改进算术优化算法求解以最小化最大完工时间为优化目标的偏序柔性车间调度模型。在算法中充分考虑解码的多约束性并通过工件前继工序和机器可用时间段确定后继工序的最优加工时间，在勘探阶段设计了基于适应度和工序完工差异度的二维聚类，并通过类间交叉和有效变异策略扩大种群多样性，在开发阶段针对瓶颈工单和瓶颈机器设计了一种等级邻域策略增强算法的局部搜索能力。通过仿真实验，采用扩展基准算例在算法比较结果和运行时间上验证了基于等级邻域策略的改进算术优化算法应用在偏序柔性车间调度的优越性。

关键词：柔性车间；等级邻域；优化算法；二维聚类；差异度
中图分类号：TP278 文献标识码：A
DOI：10.19287/j.mtmt.1005-2402.2022.07.028

Optimal scheduling of partially ordered flexible job shop based on hierarchical neighborhood strategy

LI Changyun①, LIN Duo②, HE Pinjie①, GU Pengfei②
(①School of Computing, Hunan University of Technology, Zhuzhou 412007, CHN;
②College of Rail Transit, Hunan University of Technology, Zhuzhou 412007, CHN)

Abstract: Aiming at the scheduling problem of workpiece process with complex partially ordered relationship in garment job shop, the characteristics and difficulties of this kind of problem are analyzed, and the improved arithmetic optimization algorithm is used to solve the complex partially flexible job shop scheduling model with the optimization goal of minimizing the maximum completion time. In the algorithm, the multi constraints of decoding are fully considered, and the optimal processing time of subsequent processes is determined through the previous process of the workpiece and the available time period of the machine. In the exploration stage, a two-dimensional clustering based on fitness and process completion difference is designed, and the population diversity is expanded through inter class crossover and effective mutation strategies, In the development stage, a hierarchical neighborhood strategy is designed for bottleneck work orders and bottleneck machines to enhance the local search ability of the algorithm. Through simulation experiments, extended benchmark examples are used to verify the superiority of the improved arithmetic optimization algorithm based on hierarchical neighborhood strategy in partially ordered flexible job shop scheduling.

Keywords: flexible job shop; hierarchical neighborhood; optimization algorithm; two dimensional clustering; degree of difference

柔性作业车间调度问题[1](flexible job shop scheduling problem，FJSP) 是一种研究生产资源的分配与工件工序的加工顺序的问题，具有加工设备不确定和工艺路线柔性等特性。在实际的车间生产环境中，很多工件的工艺流程具有偏序关系，可能会包含工序具有共同前继工序或者多个前继工序的情

* 国家重点研发计划资助项目（2018YFB1700200）；国家重点研发计划课题（2019YFD1101305）；湖南省重点领域研发计划课题资助项目（2019GK2133）；湖南省教育厅开放平台创新基金资助项目（18K077）

况，以服装生产车间为例[2]，在布料裁剪阶段之后可以由不同机器并行完成服装不同部位的缝制，最后再统一进行拼接和后续操作。提高工序的并行性并考虑前继工序的约束性，能够实现具有偏序工艺流程的柔性车间调度，提高工厂资源的利用效率。

国内外学者对 FJSP 的研究主要集中在机器柔性和顺序工艺流程的问题，对具有复杂偏序关系的 FJSP 研究较少。其中，谢志强等[3-4] 针对串行工序的紧密度和并行工序的并行性提出考虑后续工序的择时综合调度算法。唐红涛等[5] 针对包含并行工序集的车间提出动态触发邻域机制增强算法的局部搜索能力。刘晓平等[6] 提出了工序和设备分段编码从而实现适应工件工序可并行的解码算法，缩短了最大完工时间。徐本柱等[7] 研究了工序的并行特性，有效地缩短了产品完工时间。以上算法大都对复杂的并行工序情况考虑不全面，没有有效地解决串行工序的紧凑度和并行工序的并行性之间的关系，同时未考虑瓶颈工单和瓶颈机器中工件的偏序关系对最小化完工时间的影响。

本文针对偏序柔性作业车间调度问题 (partial order flexible job-shop scheduling problem， POFJSP) 提出了改进的算术优化算法 (improved arithmetic optimization algorithm, IAOA)，设计了基于适应度和工序完工差异度的二维聚类交叉和有效变异策略，并引入一种等级邻域变异策略对瓶颈工单和瓶颈机器上的工序分等级进行变异，有效地缩短了具有偏序关系工件的最大完工时间。最后通过仿真实验对比其他优化算法验证本文所提算法的可行性和优越性。

1 问题模型描述和约束建立

1.1 偏序柔性车间调度问题描述

FJSP 中工件工序按照顺序关系调度[8]，每个工序在加工机器集中选择一个机器进行生产。服装车间中工件的生产需要经过多个工序的加工，每个工序可由一个或多个机器进行加工，每个工序存在一个或多个前继工序。因此服装车间的调度问题和 FJSP 比较类似，但是需要考虑工件加工具有复杂偏序工艺流程的特点。故将服装车间的调度问题抽象为 POFJSP，该问题是 FJSP 的衍生，针对工件的生产工艺流程通常不是顺序型工艺流程，而是具有偏序关系的生产加工关系调度进行研究。

该问题可以采用如下描述：每台机器完成不同的工单工序所花费的时间不同，每个工单加工产品的工艺流程不同，假设共有n个工单$Job=\{J_1, J_2, J_3, \cdots, J_n\}$，每个工单$J_i(1\leqslant i\leqslant n)$包含$p_i$个工序$O=\{O_{i1}, O_{i2}, O_{i3}, \cdots, O_{ip_i}\}$，工单工序可以在$m$个机器$M=\{M_1, M_2, M_3, \cdots, M_m\}$上进行生产，若工单工序在某些机器$M_k(1\leqslant k\leqslant m)$上不能生产，则对应的生产时间为空。一个工件的工序可能具有一个或多个前继工序以及后继工序，工序O_{ij}具有前继工序集F_{ij}和后继工序集S_{ij}。工序的加工时间要介于前继工序与后继工序之间，并具有严格的加工次序约束。

工件工艺流程如图 1 所示，工件J_1的工艺流程中工序O_{12}, O_{13}, O_{14}拥有共同的前继工序O_{11}，工序O_{11}的后继工序集$S_{11}=\{O_{12}, O_{13}, O_{14}\}$可并行加工；工序$O_{26}$有多个前继工序$O_{24}, O_{25}$，其最早开始时间为前继工序集$F_{26}=\{O_{24}, O_{25}\}$中结束时间$ET_{24}, ET_{25}$较晚的值。

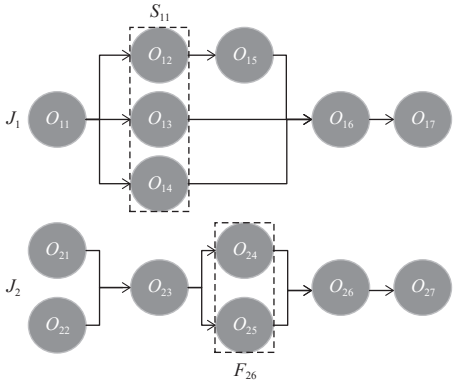

图 1 工件工艺流程图

1.2 数学模型建立

针对偏序柔性车间调度问题，本文基于以下假设建立数学模型：

（1）零时刻可加工所有工件的第一个工序，且所有机器属于待工状态。

（2）一台机器上不能同时加工多个工序。

（3）同一个工件的前继工序全部加工完成后才能开始下一道工序。

（4）某工件一旦在某机器上开始加工，不可中断当前工序的运作。

（5）工件加工互不影响，机器加工互不影响。

为了后续描述表达方便，现将各变量定义如表 1 所示。

本文研究的优化目标为最小化机器最大完工时间[9]，基于上述假设条件和数学符号描述模型约束

$$T_i=\max_{1\leqslant j\leqslant p_i}\{ET_{ij}\}\forall 1\leqslant i\leqslant n \quad (1)$$

$$MS = \min(\max_{1\leqslant i\leqslant n}\{T_i\}) \quad (2)$$

$$\sum_{j=1}^{p_i} M_{ijk} = 1 \forall 1 \leqslant i \leqslant n, k \in M \quad (3)$$

$$F_{ij} = \{O_{iq} | 1 \leqslant i \leqslant n, 1 \leqslant q < j\} \vee \emptyset \quad (4)$$

$$S_{ij} = \{O_{iq} | 1 \leqslant i \leqslant n, j < q \leqslant p_i\} \vee \emptyset \quad (5)$$

$$ST_{ij} \geqslant TST_{ij} = \max(ET_{iq})$$

$$\forall O_{iq} \in F_{ij}, 1 \leqslant i \leqslant n, 1 \leqslant q < j \quad (6)$$

$$ET_{ij} \geqslant TST_{ij} + MT_{ijk} \quad (7)$$

$$IM_{vk} = ST_{(v+1)k} - ET_{vk} \geqslant 0 \quad (8)$$

其中：式（1）表示工件J_i（$1 \leqslant i \leqslant n$）的最大完工时间；式（2）表示优化目标为最小化机器最大完工时间，即适应度；式（3）表示工件的每道工序只能在一台机器上加工；式（4）表示工件工序含有一个或者多个前继工序，规定工单开始工序的前继工序为空；式（5）表示工件的结束工序不包含后继工序（置为空集），其他工序的后继工序集中包含一个或多个工序；式（6）表示需要完成当前工单工序的全部前继工序后才开始加工；式（7）表示O_{ij}的最早结束时间为其前继工序的完成时间的最大值与其所在加工机器上的加工时间之和；式（8）表示机器加工当前工序完成之后才能开始下一个工序的加工。

表 1　符号及其描述

符号	含义
i	工件编号，$1 \leqslant i \leqslant n$
j	工序编号，$1 \leqslant j \leqslant p_i$
k	机器编号，$1 \leqslant k \leqslant m$
O_{ij}	工件i工序j
s	当前迭代中的第s个个体
ST_{ij}	O_{ij}的开始时间
ET_{ij}	O_{ij}的结束时间
p_i	工件J_i最大工序号
M_{ijk}	O_{ij}在机器k加工，可加工为1，否则为0
MT_{ijk}	O_{ij}在机器k上需要加工时间
F_{ij}	O_{ij}的前继工序集
S_{ij}	O_{ij}的后继工序集
IJ_{ij}	O_{ij}与前一个工序的间隔时间
OM_{vk}	机器k上第v工序
STM_{vk}	机器k上第v个工序的开始时间
ETM_{vk}	机器k上第v个工序的结束时间
IM_{vk}	机器k上第v个工序与后一个工序的间隔时间
TST_{ij}	O_{ij}的前继工序的最大加工结束时间
T_i	工件$J_i(1 \leqslant i \leqslant n)$的最晚完成时间
MS	机器最大完工时间

2　基于等级邻域策略的改进算术优化算法（IAOA+GNS）设计

算术优化算法 (arithmetic optimization algorithm, AOA)[10] 是一种群智能算法，用于解决连续型运筹优化问题，具有调整参数少，收敛速度快等优点，其思想适用于解决偏序柔性车间调度问题。在原始的 AOA 中，将其分为两个主要阶段：勘探和开发。在勘探阶段算法先对搜索空间进行广泛覆盖搜索，以避免陷入局部解，开发阶段对勘探阶段得到的解进行邻域搜索，算法步骤如下。

Step1：设定种群迭代次数T，种群大小$popsize$，随机初始化生成一组候选解$X(1)$。首先解码根据式（2）计算种群适应度MS，并将当代种群的最佳候选解视为目前获得的最佳解$X_{\text{best}}(t)$，然后通过式（9）计算$MOA(t)$进行阶段选择。$MOA(t)$为选择概率，Min为最小概率，Max为最大概率，本文设定$Min = 0.2$，$Max = 1$，随着迭代次数的增加，选择开发阶段概率逐渐大于勘探阶段。

$$MOA(t) = Min + t \times \left(\frac{Max - Min}{T}\right) \quad (9)$$

Step2：勘探阶段，在$X_{best}(t)$附近使用搜索策略在多个解空间上进行随机搜索，从而扩大种群多样性。

Step3：开发阶段，利用搜索策略在适应度值较高的解邻域进行深入搜索，并在经过多次迭代之后发现接近最优的解。

AOA 算法主要用于求解连续型问题，本文研究的 POFJSP 属于离散型问题，因此提出一种基于偏序工序的改进算术优化算法（IAOA），其次在勘探阶段设计了基于适应度和工序完工差异度的二维聚类交叉和有效并行变异策略增加种群多样性，在开发阶段设计了一种等级邻域搜索（grade neighborhood search，GNS）策略增强算法的局部搜索能力。最后结合两者提出基于等级邻域策略的改进算术优化算法（IAOA+GNS）求解 POFJSP。

基于等级邻域变异策略的改进算术优化算法流程如图 2 所示。

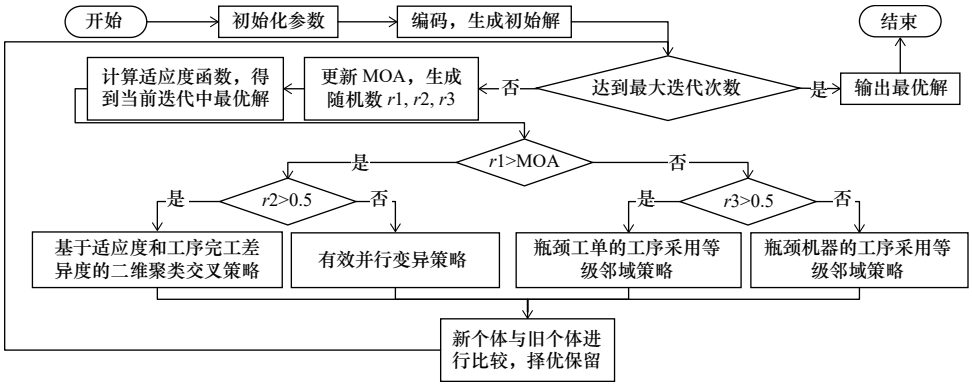

图 2 IAOA+GNS 算法流程图

2.1 编码解码

POFJSP 需要考虑到工件工序以及生产机器的安排，本文采取双编码形式$X_i(t)=[X_g|X_m]$对工序和机器进行编码，X_g为工序编码部分，工件J_i出现的次数为其第j个工序，X_m为机器编码部分，表示对应工序O_{ij}的可选机器集中的机器编号M_{ijk}。编码形式如图 3 所示。

O_{ij}	O_{11}	O_{12}	O_{13}	O_{21}	O_{22}	O_{31}	O_{32}	O_{33}
X_g	1	1	1	2	2	3	3	3
M_{ij}	M_{111}	M_{121}	M_{131}	M_{211}	M_{222}	M_{321}	M_{322}	M_{332}
X_m	1	1	1	1	2	2	2	2

图 3 编码示例图

解码是群智能算法能否解决车间调度问题的关键，考虑解码的多约束性并通过工件前继工序和机器可用时间段可以确定后继工序的最优加工时间，IAOA 进行解码时具有以下步骤：

Step1：将候选解$X_i(t)$的工序编码部分换算为工件工序O_{ij}列表，找到对应的安排机器M_{ijk}。

Step2：将前继工序为空集的工序的最早可开始时间TST_{ij}初始化为 0，其他的工序则按照式（6）计算其TST_{ij}。

Step3：对O_{ij}使用基于插入优先的解码策略：当O_{ij}安排的M_{ijk}上已安排的工序数大于 0 时，转到 Step3.1，否则转到 Step3.3。

Step3.1：判断O_{ij}能否插入到M_{ijk}上的第一个工序前，如果第一个工序的开始时间STM_{1k}大于等于O_{ij}的TST_{ij}和其加工时间MT_{ijk}之和，则跳转到 Step4，否则判断M_{ijk}上已安排的工序数是否大于 1，是则跳到 Step3.2，否则跳到 Step3.3。

Step3.2：循环安排在M_{ijk}上，且时间在O_{ij}的TST_{ij}之后的所有工序OM_{vk}，使用式（8）计算两个工序之间的时间间隔IM_{vk}，在O_{ij}的TST_{ij}和工序OM_{vk}的结束时间ETM_{vk}中选择较大的值，判断该值和O_{ij}的加工时间MT_{ijk}之和是否小于下一个工序的开始时间$STM_{(v+1)k}$，并且IM_{vk}是否大于等于O_{ij}的MT_{ijk}，是则表示可以插入该空闲时间段，转到 Step4。循环完毕，如发现没有可插入的时间段，则跳转到 Step3.3。

Step3.3：O_{ij}不能插入M_{ijk}上的空闲时间段，在O_{ij}的TST_{ij}和M_{ijk}上已安排的最后一道工序的结束时间ETM_{vk}中选择较大的值，该值即为O_{ij}的真正开始时间ST_{ij}。

Step4：利用式（1）和式（2）算出目标函数值。

如图 4 所示，有 3 个具有偏序关系的工件，将部分工序按照解码步骤解码到调度甘特图，以工序O_{22}为例，其$TST_{ij}=3$，机器M_1安排工序数大于 2，转到 Step3.2 检查空位 1 发现$IM_{12}\geqslant MT_{221}$但$MT_{221}+TST_{22}\leqslant STM_{21}$，检查空位 2 发现$IM_{22}\geqslant MT_{221}$且$ETM_{21}+MT_{221}\leqslant STM_{31}$，符合条件，可插入空位 2。

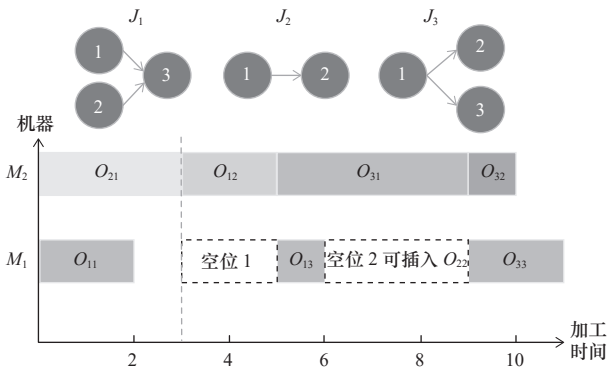

图 4 解码调度甘特图

2.2 种群初始化

种群初始化的质量好坏直接影响到算法求解速度和最终解的质量，为了减少无效解的产生并增加种群多样性，采用以下两种方法进行种群初始化（两

种方法生成的初始解在初始种群数量中各占 50%）。

正向法：首先随机打乱工序编码，并为每道工序随机分配机器，随后通过前继工序表查看所有工序，将具有共同前继工序的工序编码排在一起，最后检查同一工单中具有共同前继工序的工序的机器编码，如果在同一台机器上加工，则在可选机器集中重新选择机器进行分配。

逆向法：由正向法生成的工件编码通过倒置得到逆向编码，机器编码仍采用正向法编码规则随机生成。

2.3 勘探阶段

IAOA 算法在传统 AOA 算法上融合了 GA 算法的思想，将搜索策略替换成了交叉和变异，以此解决离散调度问题，因此在勘探阶段中使用二维聚类交叉策略和有效并行变异策略扩大种群多样性。

2.3.1 二维聚类交叉策略

本文使用聚类交叉方法进行操作，由于不同适应度值的两类个体基因存在相似可能性，因此对种群内每个个体引入工序完工差异度（degree of variance of process completion, $DVPC_s$），$DVPC_s$可以计算出不同个体基因之间的差异度，结合适应度值组成二维聚类再进行交叉增大种群多样性。工序完工差异度计算式（10）如下。

$$DVPC_s = \sum_{i=1}^{n}\sum_{j=1}^{p_i}\left|ET_{ij} - E\hat{T}_{ij}\right| \qquad (10)$$

其中：$E\hat{T}_{ij}$为当前迭代次数中适应度最好的个体X_{best}的所有工单工序的结束时间。

首先以当前迭代种群最优个体X_{best}作为基准，对种群中所有个体的适应度MS_s和$DVPC_s$统一进行最大最小归一化，范围为 [0, 1]，并以MS_s和$DVPC_s$作为指标绘制散点图。随后使用 k-means 聚类算法对种群进行二维聚类，其中参数 k 为聚类的数量，本文将 k 设为 4，对应不同MS_s和$DVPC_s$下的个体类，聚类算法结果如图 5 所示。

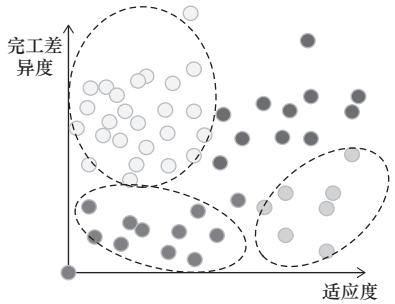

图 5 适应度和完工差异度聚类图

随后随机选择两类中的个体作为父代，保留其中适应度较小的父代中的优秀基因，优秀基因为个体基因中工单紧凑度（work order compactness，WOC）高的工单，WOC 可通过比较工单理想完成时间和工单实际完成时间获得，如式（11）所示。

$$WOC_i = \sum_{j=1}^{p_i} MT_{ijk} \Big/ ET_{ip_i} \qquad (11)$$

优秀基因的保留能够加快迭代速度并优化子代基因以获取更优解。最后使用 POX 交叉将父代P_1的优秀基因保留，其他基因删除，并将另一父代P_2中除了优秀基因工单外的其他工单从左到右依次填补到P_1的空基因位上得到交叉后的子代基因。

2.3.2 有效并行变异策略

针对变异过程中可能存在的无效变异问题，本文对机器变异和工序变异进行了以下改进。

机器变异：随机选择总数量中 20% 的工序，将其对应机器换成可选机器集中加工时间更短的机器进行加工。

工序变异：针对不同机器上的工序进行位置交换可能对该机器的加工工序数量和顺序不具有任何影响，从而导致无效变异的问题。本文对同一台机器上加工的任意多个工序交换位置，可有效减少无效变异。

2.4 开发阶段

影响适应度函数的直观原因在于以下两点：一是瓶颈工单的 WOC 在所有工单的 WOC 中处于较低值，大部分工序在对应机器上的加工时间比较长，且与前继工序之间的时间间隔较大；二是瓶颈机器上的利用率远高于其他机器，瓶颈机器上加工的工序将对目标函数值起到决定作用。

基于上述问题，开发阶段在多个密集解空间上深入搜索最优解，使用等级邻域搜索策略（GNS）分别对瓶颈工单和瓶颈机器两个方面进行邻域搜索。GNS 策略如下：

（1）优先级设定。首先为不同情况下的工序设定优先等级，具有多个后继工序的工序在机器上的完成时间能够影响到所有后继并行工序的完成进度，是缩短工单的最大完工时间的重要因素，因此该类工序为最高优先级 G1；多数工序只存在一个后继工序，可以通过对该类工序使用优化策略缩小目标函数值，因此设为中等优先级 G2；最后一道

工序对整个工单的完成时间影响不大，设为最低优先级 G3。

（2）变异策略设定。针对不同原因对工序进行不同的变异策略，分为以下 3 种情况讨论。

GNS1：如果当前工序的开始时间和前继工序的完成时间间隔大于当前工序加工时间的一半，则将当前工序与对应机器上其前继工序完成时间之后开始加工的其他工序随机进行位置互换；

GNS2：如果当前工序与前继工序的时间间隔较小，则将当前工序换到可选机器集中加工时间更少的机器；

GNS3：如果当前工序所处机器上没有空闲时间段，则将当前工序交换至可选机器集中机器负载更小的机器上。

若同时满足多种情况，则随机选择一种情况进行变异。

（3）瓶颈工单或瓶颈机器变异。对瓶颈工单或瓶颈机器中的所有工序进行优先级排序，随后使用轮盘赌算法按照优先级随机挑选 10% 工序，并根据不同情况进行变异。

以图 4 所示的解码调度甘特图为例，假设在第t次迭代中对瓶颈工件J_3进行等级邻域变异，其中O_{31}的优先级最高有多个后继工序$S_{31}=\{O_{32},O_{33}\}$，使用 GNS3 策略选择可选机器集中负载更低的机器M_1进行加工，得到的调度甘特图如图 6 所示，有效地提前了工件J_3的部分工序加工时间。

假设在第$t+1$次迭代中对瓶颈机器M_1上的加工工序进行等级邻域变异，采用 GNS1 策略对工序O_{22}与其前继工序的完成时间之后开始加工的工序O_{13}进行位置交换，调度甘特图如图 7 所示，最大完工时间从原先的 11 有效地缩短到 10。

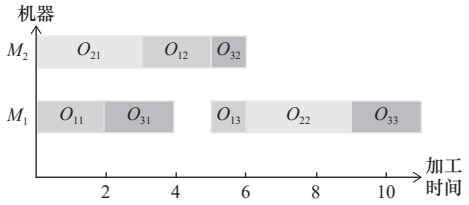

图 6 瓶颈工件使用等级邻域策略结果图

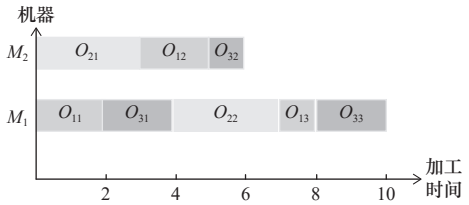

图 7 瓶颈机器使用等级邻域策略结果图

3 仿真实验与案例分析

目前没有标准算例验证本文的研究问题，因此本文基于 Brandimarte P[11] 所提出的 10 个案例 Mk01~Mk10 拓展了适用于偏序柔性车间调度问题的测试算例，表 2 为生成的扩展算例的参数，新算例 PMk01~PMk10 在基准算例的基础上添加了每个工序的前继工序集合，为了表达方便，假设每个算例中工件的工艺流程一样。

3.1 参数对比实验

智能优化算法的优化结果受算法参数的影响很大，影响算法性能的参数有：种群规模*popsize*，最大迭代次数T，本文实验所用算法均采用相同参数

表 2 扩展算例参数

新算例	工件数	机器数	总工序数	前继工序集
PMk01	10	6	55	∅,{1},{1},{1},{2,3},{4,5}
PMk02	10	6	58	∅,{1},{1},{1},{2,3},{4,5}
PMk03	15	8	150	∅,{1},{1},{1},{2},{3},{4},{5},{7},{6,8,9}
PMk04	15	8	90	∅,{1},{1},{1},{2,3},{4},{5},{6,7},{8}
PMk05	15	8	106	∅,{1},{1},{1},{2,3},{4},{5},{6,7},{8}
PMk06	15	4	150	∅,{1},{1},{1},{2},{3},{4},{5},{7},{8},{6,9,10},{11},{12},{13},{14}
PMk07	10	15	100	∅,{1},{1},{1},{2,3,4}
PMk08	20	5	225	∅,{1},{1},{1},{2},{3},{4},{5},{7},{8},{6,9,10},{11},{12},{13}
PMk09	20	10	240	∅,{1},{1},{1},{2},{3},{4},{5},{7},{8},{6,9,10},{11},{12},{13}
PMk10	20	15	240	∅,{1},{1},{1},{2},{3},{4},{5},{7},{8},{6,9,10},{11},{12},{13}

值。以 PMk09 为测试算例，选择规模为 (3^2) 的正交实验进行测试，为避免结果的随机性，以 30 次结果的平均值作为评价指标，正交实验结果如表 3 所示。

表 3　正交实验结果比较表

实验编号	*popsize*	*T*	均值 AVR_{30}/h
1	80	50	316.86
2	90	50	312.66
3	100	50	310.95
4	80	60	310.26
5	90	60	307.69
6	100	60	308.91
7	80	70	309.34
8	90	70	307.61
9	100	70	308.43

可以从表中看出当种群规模 $popsize = 90$，最大迭代次数 $T = 70$ 时算法的性能最好，但是当群规模 $popsize = 90$，最大迭代次数 $T = 60$ 时算法效果已经接近最优性能，且运行时间少于达到最优性能所花费的时间，所以本文选择群规模 $popsize = 90$，最大迭代次数 $T = 60$ 作为 PMk09 算例的最优参数。

3.2　算法比较及结果

为了证明本文的 IAOA+GNS 算法应用在偏序柔性车间调度上的有效性，将本文所采用的方法与文献 [12] 和文献 [13] 的算法求解效率进行对比，对比算法的参数与本文算法的一致。同时为了验证本文等级邻域策略的有效性，选择与去掉 GNS 策略采用传统变邻域搜索的 IAOA+VNS 算法进行对比，其中 IAOA+VNS 算法参数与 IAOA+GNS 一致。仿真软件使用 MATLAB 2016a，电脑硬件参数为：处理器 Intel(R) Core(TM) i5-4210U CPU @ 3.70 GHz，内存 8 GB。算法各运行 30 次，并使用运行结果中的最小完工时间 R_{best}、平均完工时间 R_{avr}、完工时间标准差 R_σ 和算法平均运行时间评测算法性能。

根据表 4 的仿真结果可以分析出本文的 IAOA+GNS 算法应用在偏序柔性车间调度具有较好的效果，最小完工时间和平均完工时间均优于其他优化算法，且完工时间标准差少于其他算法，说明本文算法能够得到更优解，且更稳定性。同时基于 IAOA+GNS 与 IAOA+VNS 算法的对比试验可以说明 GNS 策略的有效性。

根据表 5 的算法平均运行时间来看，算例 PMk05、PMk07、PMk09 在本文的运行时间比其他算法运行时间更短，最小化最大完工时间也更短。在 PMk08 算例中大部分工序只能在同一台机器上进行加工，其适应度函数取决于在此机器上加工的工序完成时间，因此无法体现本文算法的优越性。在其他算例中本文算法比其他优化算法的平均运行时间更长，但是本文算法的运行结果更优，且运行时间在可允许范围之内。同时，IAOA+VNS 算法比 IAOA+GNS 算法运行时间少很多，但是其算法稳定性相较于较差，体现了本文改进的 IAOA+GNS 算法在稳定性上的优势。综上所述，本文算法在解决偏序柔性车间调度问题上相较于现有其他几个算法在性能和稳定性上更优越。

表 4　算法效果比较

算例名称	IAOA+GNS				IAOA+VNS			文献[12]			文献[13]		
	R_{best}	R_{avr}	R_σ	群规模	R_{best}	R_{avr}	R_σ	R_{best}	R_{avr}	R_σ	R_{best}	R_{avr}	R_σ
PMk01	38	39.62	1.694	40	40	45.43	7.264	41	42.3	2.419	40	41.2	2.981
PMk02	26	28.56	2.866	110	28	31.56	3.864	28	31.6	3.438	27	27.9	2.922
PMk03	204	204	0	30	204	210.61	7.953	204	204.0	0	204	204	0
PMk04	60	60.54	0.627	60	64	66.68	3.984	65	68.8	4.688	65	66.4	3.399
PMk05	172	175.52	3.634	70	176	178.52	2.691	173	176.5	5.359	176	176.8	1.697
PMk06	58	61.25	2.364	60	61	66.25	6.368	73	79.6	11.378	71	71.4	1.957
PMk07	139	142.36	4.398	90	154	158.75	4.126	146	150.5	6.357	144	145.6	3.322
PMk08	523	523.00	0	10	523	523	0	523	523.0	0	523	523.6	1.369
PMk09	307	307.43	1.236	80	311	316.65	6.685	319	325.2	9.674	316	323.6	9.658
PMk10	206	224.65	19.369	80	239	251.65	13.384	248	257.2	13.397	238	242.4	5.389

表 5 算法平均运行时间

CPU/s	算例									
	PMk01	PMk02	PMk03	PMk04	PMk05	PMk06	PMk07	PMk08	PMk09	PMk10
IAOA+GNS	24.09	36.08	35.94	56.66	102.32	104.39	123.65	5.36	286.31	243.36
IAOA+VNS	20.96	55.01	28.22	48.29	91.34	85.93	82.23	4.43	204.37	215.06
文献[12]	25.22	36.32	36.29	67.36	165.89	72.65	151.24	6.79	321.23	231.39
文献[13]	26.36	38.36	40.36	59.36	165.25	62.65	164.25	5.38	305.25	220.36

以算例 PMk09 为例，图 8 展示了本文算法对该算例的调度甘特图，从图中可以看出，其最大完成时间为 307，8 号机器为瓶颈机器，该机器上已无空闲时间可进行调度，限制了适应度函数的最小值。

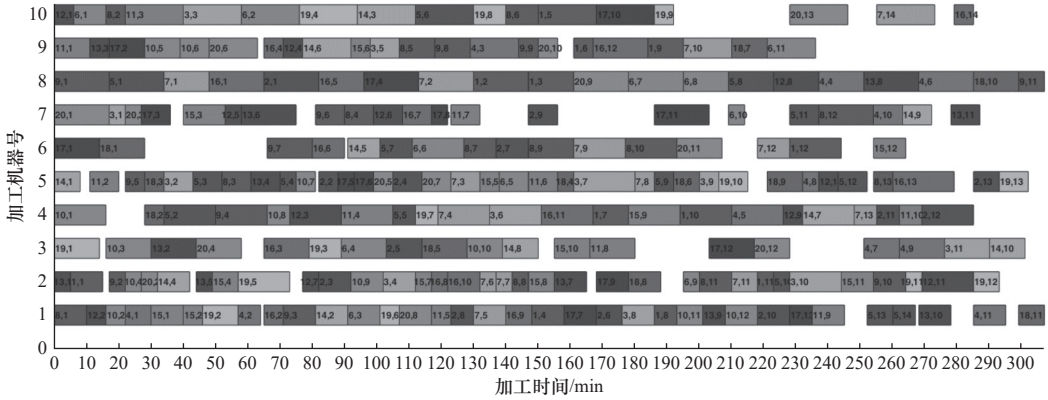

图 8 算例 PMK09 调度甘特图

4 结语

针对服装加工行业中工件具有复杂工艺流程的情况，本文提出了 IAOA+GNS 算法求解该类偏序柔性车间调度问题。算法充分考虑了前继工序的约束性，首先基于适应度和工序完工差异度设计出聚类交叉和有效变异策略，避免无效变异的同时扩大了种群探索范围；然后利用瓶颈机器和瓶颈工件的关键性进行等级邻域变异以搜索更优解。通过仿真实验对比其他算法，验证了 IAOA+GNS 算法的有效性和稳定的求解性能。接下来将下一步的研究重点放到动态调度偏序柔性车间调度上。

参 考 文 献

[1] 梁晓磊, 马千慧, 李章洪, 等. 考虑多时间约束和机器效率的柔性作业车间调度问题建模及优化[J]. 制造技术与机床, 2021(10): 114-122.
[2] 沈海娜, 邵一兵, 陈才芽. 基于编码技术的服装工艺单设计系统开发与应用[J]. 纺织导报, 2021(3): 30-33.
[3] 谢志强, 张晓欢, 高一龙, 等. 考虑串行工序紧密度的择时综合调度算法[J]. 机械工程学报, 2018, 54(6): 191-202.
[4] 谢志强, 张晓欢, 辛宇, 等. 考虑后续工序的择时综合调度算法[J]. 自动化学报, 2018, 44(2): 344-362.
[5] 唐红涛, 杨志鹏. 基于工序集的多阶段混合流水车间调度问题研究[J]. 工业工程与管理, 2021, 26(4): 60-68.
[6] 刘晓平, 徐本柱, 彭军, 等. 工件工序可并行的作业车间调度模型与求解[J]. 计算机辅助设计与图形学学报, 2012, 24(1): 120-127.
[7] 徐本柱, 费晓璐, 章兴玲. 柔性作业车间批量划分与并行调度优化[J]. 计算机集成制造系统, 2016, 22(8): 1953-1964.
[8] 杨俊茹, 李贺, 李瑞川, 等. 基于NSGA-Ⅱ算法的液压元件多目标柔性作业调度研究[J]. 制造技术与机床, 2021(6): 112-116.
[9] 唐浩, 黎向锋, 张丽萍, 等. 扰动机制下的遗传算法求解柔性作业车间调度[J]. 现代制造工程, 2021(7): 1-9,37.
[10] Laith A, Ali D, Seyedali M, et al. The arithmetic optimization algorithm[J]. ScienceDirect, 2021, 376: 113609.
[11] Brandimarte P. Routing and scheduling in a flexible job shop by tabu search[J]. Annals of Operations Research, 1993, 41(3): 157-183.
[12] 刘博, 袁欣, 明新国. 基于遗传算法和变邻域搜索的柔性作业车间同步调度方法[J]. 机械设计与研究, 2021, 37(1): 177-182,189.
[13] 顾幸生, 丁豪杰. 面向柔性作业车间调度问题的改进博弈粒子群算法[J]. 同济大学学报:自然科学版, 2020, 48(12): 1782-1789.

第一作者：李长云，男，1971 年生，博士，教授，研究方向为软件自动化、物联网应用及计算机工业控制等，SCI/EI 收录 10 余篇。E-mail: lcy469@163.com

通信作者：林多，女，1998 年生，硕士，研究方向为车间调度。E-mail: 13272318556@163.com

（编辑 高 扬）

（收稿日期：2021-03-11）

文章编号:20220728

如果您想发表对本文的看法，请将文章编号填入读者意见调查表中的相应位置。