

# 设备带有恶化特性的作业车间调度模型与算法

黄敏<sup>1,2</sup> 付亚平<sup>1,2</sup> 王洪峰<sup>1,2</sup> 朱兵虎<sup>1,2</sup> 王兴伟<sup>1,2</sup>

**摘要** 考虑到现实作业车间调度中设备具有恶化特性, 针对作业的处理时间是开始时间的线性递增函数的作业车间调度问题, 建立了以最小化最迟完成时间为目标的优化模型, 进而设计了嵌套分割算法进行求解. 该算法在抽样阶段嵌入单亲遗传算法以提高抽样的多样性和质量. 实例结果表明, 所提出的算法在解决该问题上可以获得较高质量的解, 并且具有很好的鲁棒性.

**关键词** 嵌套分割算法, 单亲遗传算法, 作业车间, 设备恶化, 调度问题

**引用格式** 黄敏, 付亚平, 王洪峰, 朱兵虎, 王兴伟. 设备带有恶化特性的作业车间调度模型与算法. 自动化学报, 2015, 41(3): 551–558

**DOI** 10.16383/j.aas.2015.c131067

## Job-shop Scheduling Model and Algorithm with Machine Deterioration

HUANG Min<sup>1,2</sup> FU Ya-Ping<sup>1,2</sup> WANG Hong-Feng<sup>1,2</sup> ZHU Bing-Hu<sup>1,2</sup> WANG Xing-Wei<sup>1,2</sup>

**Abstract** For the job-shop scheduling problem, a job-shop scheduling model with machine deterioration is built in order to minimize makespan, considering that the processing time of jobs is a linearly increasing function of the start time. Then a nested partition method is designed for solving it. In the sampling process, the partheno-genetic algorithm is embedded into the nested partition method in order to ensure the diversity of sampling and quality. Simulation experiments show that the proposed algorithm for solving job-shop scheduling problem with machine deterioration can get higher quality solutions and have a better robustness.

**Key words** Nested partition method, partheno-genetic algorithm, job shop, machine deterioration, scheduling problem

**Citation** Huang Min, Fu Ya-Ping, Wang Hong-Feng, Zhu Bing-Hu, Wang Xing-Wei. Job-shop scheduling model and algorithm with machine deterioration. *Acta Automatica Sinica*, 2015, 41(3): 551–558

调度是影响制造型企业生产效率的关键因素, 建立合理的调度模型及寻找有效的调度方法和优化技术是提高制造型企业生产效率、降低生产成本的重要途径. 作业车间调度问题 (Job-shop scheduling problem, JSSP) 是众多制造型企业普遍存在的生产调度问题, 其蕴含着复杂的生产制约关系, 既要考虑

作业加工的先后次序关系, 又要考虑加工进程的协调, 是一类复杂的组合优化问题.

众多学者对作业车间调度问题进行了深入而系统的研究<sup>[1]</sup>. 传统的作业车间调度的研究假设作业处理时间为可知的常数<sup>[2]</sup>. 而在实际加工过程中, 处理时间受到诸多不确定因素的影响, 如文献 [3–4] 考虑了作业处理时间服从三角模糊数的不确定作业车间调度问题, 文献 [5–6] 分别研究了作业的处理时间服从均匀分布和正态分布的作业车间调度问题. 上述研究考虑作业的处理时间预先确定、模糊或随机的作业车间调度问题, 但均未考虑由于设备工作效率降低或工件物理特性变化而引起的处理时间延长的问题.

20 世纪 90 年代, Gupta 等<sup>[7]</sup> 考虑到工件特性、设备磨损及操作者疲惫程度的影响, 假设工件的处理时间是其开始时间的线性递增函数, 首先提出了工件具有恶化特点的调度模型. 这种考虑恶化的调度模型弥补了传统调度中模型缺乏实用性的不足, 且具有广泛的应用背景, 如钢铁生产、清洁维护及服务业务等. 目前, 考虑恶化情形的调度问题的研究多以单机<sup>[8–10]</sup> 和流水线<sup>[11–12]</sup> 为背景, 而在作业车间背景下的研究较少<sup>[13]</sup>. Mosheiov<sup>[14]</sup> 首次对具有恶化时

收稿日期 2013-11-19 录用日期 2014-10-27  
Manuscript received November 19, 2013; accepted October 27, 2014

国家杰出青年科学基金 (71325002, 61225012), 国家自然科学基金 (71071028, 71001018), 流程工业综合自动化国家重点实验室基础科研业务费 (2013ZCX11), 中央高校基本科研业务费专项基金 (N130404017) 资助

Supported by National Science Foundation for Distinguished Young Scholars of China (71325002, 61225012), National Natural Science Foundation of China (71071028, 71001018), Fundamental Research Funds for State Key Laboratory of Synthetical Automation for Process Industries (2013ZCX11), and Fundamental Research Funds for the Central Universities (N130404017)

本文责任编辑 李乐飞

Recommended by Associate Editor LI Le-Fei

1. 东北大学信息科学与工程学院 沈阳 110819 2. 流程工业综合自动化国家重点实验室 (东北大学) 沈阳 110819

1. College of Information Science and Engineering, Northeastern University, Shenyang 110819 2. State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819

间的作业车间调度问题进行了复杂性研究,指出在实际处理时间与其开始时间呈线性关系的情况下,最小化最迟完成时间的作业车间调度问题是 NP-complete 问题. 文献 [13, 15] 研究带有恶化情形下的批量作业车间调度问题,假设作业的处理时间与其开始时间呈指数关系. 文献 [16] 考虑了作业的处理时间具有线性恶化情形的柔性作业车间调度问题. 现有对恶化特性的作业车间调度问题的研究,均假设作业在所有设备上的恶化系数相同. 而在有些实际生产环境中,如带有自动和手控设备的作业车间,作业的恶化系数往往受到设备磨损、操作者疲惫程度及工序的复杂程度的影响,因此,各作业的不同工序往往具有不同的恶化系数.

考虑到带有恶化情形的作业车间调度问题具有广泛的应用背景,本文针对具有自动和手控设备的作业车间,结合手控设备操作者的工作效率随其工作时间增加而降低的特点,研究了设备带有恶化特性的作业车间调度问题 (Job-shop scheduling problem with machine deterioration, JSSP-MD). 考虑作业的不同工序具有不同的恶化系数,并假设各工序的实际处理时间为其在相应设备上开始时间的线性函数,建立了以最小化最迟完成时间为目标的调度模型,以最大化设备利用率. Garey 等<sup>[17]</sup> 已证明 JSSP 问题属于 NP-hard 问题, JSSP-MD 是 JSSP 问题的扩展,具有更高的复杂性,因此, JSSP-MD 也属于 NP-hard 问题. 为此,本文在嵌套分割算法 (Nested partition method, NPM) 的抽样阶段嵌入单亲遗传算法 (Partheno-genetic algorithm, PGA) 的混合算法 (Nested partition method embedded partheno-genetic algorithm, NP-PGA) 进行求解. 通过求解实例,验证和分析了该算法在解决 JSSP-MD 问题的执行效果.

## 1 问题描述与模型

资源的有效利用对企业的管理决策是至关重要的,如何提高设备利用率、最小化最迟完成时间是衡量现实生产调度优化效果的重要指标. 考虑最小化最迟完成时间和恶化时间的作业车间调度问题可以描述如下:  $n$  个作业在  $m$  台设备上加工, 设  $J$  为待加工的作业集合,  $M$  为设备集合. 集合  $J$  中的每个作业  $j$  ( $j = 1, 2, \dots, n$ ,  $n$  表示作业总数) 须按照预先确定的工艺顺序加工,  $A_j = \{O_{jm_1}, O_{jm_2}, \dots, O_{jm_{K_j}}\}$ , 其中  $A_j$  表示作业  $j$  的工艺路线,  $O_{jm_k}$  表示作业  $j$  的第  $k$  个工序在设备  $m_k$  上加工,  $k = 1, 2, 3, \dots, K_j$ ,  $K_j$  为作业  $j$  的工序总数.  $S_{ji}$  和  $C_{ji}$  表示作业  $j$  在设备  $i$  上的开始处理时间和完成时间.  $p_{ji}$  为作业  $j$  在设备  $i$  上的正常处理时间,作业在处理过程中,作业的实际处理时间会由于设备磨损及

操作人员逐渐疲惫而增加,即作业的处理时间出现恶化现象. 设  $\alpha_{ji}$  为设备  $i$  加工作业  $j$  的恶化系数,假设恶化时间与其对应工序的开始时间呈线性关系,则作业  $j$  在设备  $i$  上的实际处理时间  $p'_{ji}$  如式 (1) 所示,其中  $M_1$  和  $M_2$  分别表示不具有恶化特性的设备集和具有恶化特性的设备集. 调度目标是确定各台设备上作业的加工顺序,以最小化最迟完成时间,并且满足如下约束条件:同一时刻同一设备只能处理一个作业;作业的加工不允许中断;不同作业的工序之间没有先后约束;所有设备在 0 时刻均可用.

$$p'_{ji} = \begin{cases} p_{ji}, & \forall j \in J, \forall i \in M_1 \\ p_{ji} + \alpha_{ji}S_{ji}, & \forall j \in J, \forall i \in M_2 \end{cases} \quad (1)$$

根据上述问题描述和符号定义,对 JSSP-MD 问题建立模型如下:

$$\min \max_{1 \leq i \leq m} \left\{ \max_{1 \leq j \leq n} C_{ji} \right\} \quad (2)$$

$$C_{ji} = S_{ji} + p'_{ji}, \quad \forall j \in J, \forall i \in M \quad (3)$$

$$C_{ji} \leq S_{li} \vee C_{li} \leq S_{ji}, \quad j, l \in J, i \in M \quad (4)$$

$$C_{ji} \leq S_{jk}, \quad \forall (o_{ji}, o_{jk}), \forall A_j, j \in J, i, k \in M \quad (5)$$

$$S_{ji}, C_{ji} \geq 0, \quad \forall j \in J, \forall i \in M \quad (6)$$

$$p'_{ji} = \begin{cases} p_{ji}, & \forall j \in J, \forall i \in M_1 \\ p_{ji} + \alpha_{ji}S_{ji}, & \forall j \in J, \forall i \in M_2 \end{cases} \quad (7)$$

模型中,式 (2) 为目标函数,表示最小化最迟完成时间;式 (3) 表示作业的各工序的完成时间为开始时间与实际处理时间和;式 (4) 和式 (5) 表示由工艺约束条件决定的各作业的各工序的先后加工顺序,以及加工各个作业的设备的先后顺序;式 (6) 为变量的取值约束;式 (7) 为作业的实际处理时间和正常处理时间的函数关系式.

## 2 NP-PGA 算法

NP 算法是一种能够解决复杂的确定型和随机型优化问题的优化方法,其已被证明能够以概率 1 收敛到全局最优解. Shi 等已将 NP 算法用于求解 TSP 问题、供应链网络优化、产品设计、资源分配等领域<sup>[18-19]</sup>,并取得了很好的效果.

### 2.1 NP 算法的基本思想

设  $X$  为优化问题  $P$  的可行域空间,通过分割策略得到的区域称为可行域,各个可行域互不相交,且并集为整个可行域  $X$ . 对于离散问题,只含一个解的可行域称为单解域. 如果可行域  $\sigma \in X$  是通过分割可行域  $\eta \in X$  得到的,则称  $\sigma$  为  $\eta$  的子域,  $\eta$  为

$\sigma$  的父域. 由初始可行域到达一个可行域所分割的次数称为该可行域的深度, 初始可行域的深度为 0, 单解域具有最大的深度, 故又称为最大深度域.

NP 算法包括四个基本算子<sup>[18]</sup>: 分割 (Partition)、抽样 (Sampling)、选区 (Selection) 和回溯 (Backtracking). 其基本思想是: 1) 在算法的第  $k$  次迭代中, 如果认为  $\sigma(k) \in X$  是包含  $x^*$  的最可能域 (The most promising region), 则利用分割算子将  $\sigma(k)$  分割为  $M$  个子域, 并将可行域  $X \setminus \sigma(k)$  称为裙域 (Surrounding region), 得到  $M + 1$  个互不相交的可行域; 2) 对每个可行域  $\sigma_j(k), j = 1, 2, 3, \dots, M + 1$ , 利用抽样算子随机抽取  $N_j$  个点  $x_1^j, x_2^j, x_3^j, \dots, x_{N_j}^j, j = 1, 2, 3, \dots, M + 1$ . 计算相应的目标函数值  $f(x_1^j), f(x_2^j), f(x_3^j), \dots, f(x_{N_j}^j)$ , 并选择最好值作为该可行域的可能性指数 (Promising index)  $I(\sigma_j)$ ; 3) 基于抽取的样本, 利用选区算子确定第  $k + 1$  次迭代的最可能域. 依据此方法继续分割, 直到获得不可分割的单解域; 4) 如果最可能域为裙域  $X \setminus \sigma(k)$ , 则利用回溯算子回溯至上次迭代的最可能域, 并重新执行 1) ~ 3).

### 2.2 NP-PGA 算法求解 JSSP-MD 问题

回溯次数过多会影响到 NP 算法的执行效率, 为减少回溯次数, 必须提高抽样算子抽取的样本质量, 以保证算法沿着正确的方向搜索. 为此, 本文提出了 NP-PGA 算法, 以提高 NP 算法的搜索效率. 图 1 为 NP-PGA 算法的流程图. 为有效地求解 JSSP-MD 问题, NP-PGA 算法的基本算子设计方法如下.

分割: 假设  $n$  个作业在  $m$  台设备上加工, 已知各个作业工艺路线可确定各设备上加工的作业集合. 一个完整解分成  $m$  段, 如图 2 所示,  $\pi_j$  表示第  $j$  台设备上各作业相应工序的排序. 分割算子每次确定一个作业在一台设备上的位置, 依次确定第 1 台至第  $m$  台设备上作业的排序. 以 2 台设备 3 个作业为例, 工艺路线分别为:  $A_1 = \{O_{11}, O_{12}\}, A_2 = \{O_{22}, O_{21}\}, A_3 = \{O_{31}, O_{32}\}$ , 其执行方法如图 3 所示. 图中 “\*” 表示待分配作业的位置, 第 1 层表示两台设备均处于待分配; 第 2 层中第 1 个和第 3 个节点分别表示将作业 1 和作业 3 的第 1 个工序分配到第 1 台设备的第 1 个位置, 第 2 个节点表示将作业 2 的第 2 个工序分配到第 1 台设备的第 1 个位置; 第 3 层的两个衍生节点分别表示将作业 1 和作业 3 的第 2 个工序分配给第 1 台设备的第 2 个位置, 以此类推, 完成两台设备上的作业分配.

抽样: 嵌入 PGA 的抽样方法描述如下. 1) 从可行域中随机抽取  $N$  个解作为 PGA 的初始种群; 2) 对每一个体随机选择若干台未确定作业顺序的设

备, 随机地选择基因换位、基因段移位和基因段逆转操作, 改变相应设备的作业的排序; 3) 计算个体的目标函数值, 并取其倒数作为个体的适应度值; 4) 采用精英保留和轮盘赌方法选择下一代种群; 5) 重复执行 2) ~ 4), 直到满足停止条件, 输出最优个体, 并将该个体的目标值作为该可行域的可能性指数. 重复上述操作直至完成对所有可行域的抽样操作. 采用该抽样方法能够获得代表各个可行域质量的较好解, 从而保证算法沿着正确的方向搜索.

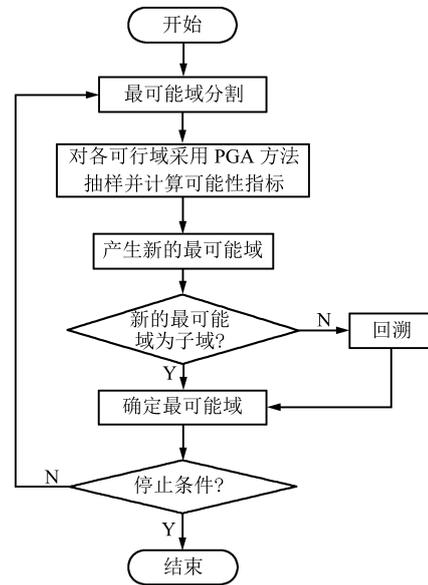


图 1 NP-PGA 算法的流程图  
Fig. 1 Graph of NP-PGA algorithm

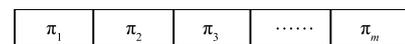


图 2 解的表达方式  
Fig. 2 Representation of solution

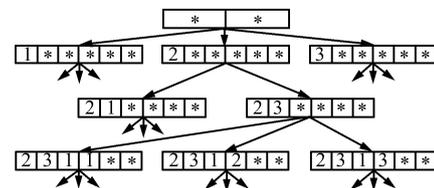


图 3 可行域的分割方法  
Fig. 3 Partition method of feasible region

选区: 如果当前迭代中具有最优的可能性指数的可行域为当前最可能域的子域, 则选取该子域为下次迭代的最可能域.

回溯: 如果当前迭代中具有最优的可能性指数的可行域为裙域, 则需要回溯操作. 本文采取两种回溯策略, 分别为: 1) 回溯到当前最可能域的父

域; 2) 回溯到截至目前找到的最好解所在可行域的父亲域。

根据上述算法设计, NP-PGA 算法执行的伪码如图 4 所示。在完成算法参数及最可能域的初始化后, 算法进入如下迭代过程: 首先将最可能域分割为一定数量的子域并构造当前裙域; 然后采用 PGA 方法对各可行域抽样, 并计算各可行域的可能性指数; 选择具有最好可能性指数的可行域, 并依据其与当前最可能域的包含关系确定是否采取回溯策略; 最后确定最可能域, 并进入下一次迭代。

### 3 实验分析

为验证算法的求解效果, 本文从某加工企业选取三个不同规模的实例进行实验, 计算机配置为 Core2 Duo 2.4 GHz CPU, 2 G RAM. 实例 1 中有 6 个作业、6 台设备; 实例 2 中有 7 个作业、7 台设备; 实例 3 中有 8 个作业、8 台设备, 实验数据分别如表 1~3 所示。表中  $p_{ji}$  和  $\alpha_{ji}$  分别表示作业  $j$  在工艺路线  $A_j$  下各工序在相应设备  $i$  上的正常处理时间和恶化系数。每一实例分别运行 30 次, 并分别取其最好值 (Best)、最差值 (Worst)、平均值 (Mean)、标准方差 (S) 及平均 CPU 时间 (Time, 单位 s) 为评价指标。实验中 NP 算法采取两个停止条件: 1) 获得单解域; 2) 最大分割次数, 本文取  $mn^2$ 。

实验首先以实例 1 为例, 验证不同回溯策略、抽样个数及 PGA 迭代次数对算法性能的影响。实验的结果及分析如下。

```

Procedure Algorithm of NP-PGA
begin
  initialize algorithm parameters;
   $t = 0$ ;
  set the whole feasible solution space as the current
  most promising region;
  repeat
    partition the most promising region  $\sigma(t)$  into mul-
    tiple subregions, and construct the surrounding
    region;
    generate the sampling solutions using PGA for
    each region;
    calculate the promising index for each region;
    select the region with the best promising index;
    identify the most promising region  $\sigma(t + 1)$ , if
    needed, the backtracking is used;
     $t = t + 1$ ;
  until a stop condition is met.
end
    
```

图 4 NP-PGA 算法伪码图

Fig. 4 Pseudo-code of NP-PGA algorithm

表 1 实例 1 的实验数据

Table 1 Experimental data of the first instance

$J$	$A_j$	$p_{ji}$	$\alpha_{ji}$
1	{ $O_{11}, O_{14}, O_{12}, O_{13}, O_{16}$ }	{8, 2, 9, 4, 17}	{0.4, 0.6, 0, 0.1, 0.4}
2	{ $O_{23}, O_{26}, O_{25}, O_{21}, O_{22}$ }	{11, 1, 9, 7, 8}	{0.1, 0.4, 0.2, 0.4, 0}
3	{ $O_{33}, O_{31}, O_{32}, O_{34}, O_{35}$ }	{17, 10, 7, 20, 9}	{0.1, 0.4, 0, 0.6, 0.2}
4	{ $O_{44}, O_{42}, O_{45}, O_{43}, O_{41}$ }	{1, 9, 16, 18, 2}	{0.6, 0, 0.2, 0.1, 0.4}
5	{ $O_{51}, O_{56}, O_{52}, O_{55}, O_{54}$ }	{2, 3, 3, 10, 11}	{0.4, 0.4, 0, 0.2, 0.6}
6	{ $O_{62}, O_{66}, O_{65}, O_{64}, O_{63}$ }	{3, 17, 13, 4, 14}	{0, 0.4, 0.2, 0.6, 0.1}

表 2 实例 2 的实验数据

Table 2 Experimental data of the second instance

$J$	$A_j$	$p_{ji}$	$\alpha_{ji}$
1	{ $O_{15}, O_{13}, O_{14}, O_{16}, O_{17}, O_{12}, O_{11}$ }	{19, 18, 7, 18, 3, 10, 15}	{0.2, 0.1, 0.6, 0.4, 0.3, 0, 0.4}
2	{ $O_{21}, O_{24}, O_{23}, O_{22}, O_{27}, O_{25}, O_{26}$ }	{14, 11, 12, 5, 2, 20, 14}	{0.4, 0.1, 0.6, 0.4, 0.3, 0, 0.4}
3	{ $O_{35}, O_{36}, O_{37}, O_{33}, O_{34}, O_{31}, O_{32}$ }	{13, 2, 4, 10, 16, 9, 8}	{0.2, 0.4, 0.3, 0.1, 0.6, 0.4, 0}
4	{ $O_{41}, O_{44}, O_{47}, O_{46}, O_{45}, O_{43}, O_{42}$ }	{7, 16, 4, 1, 9, 4, 9}	{0.4, 0.6, 0.3, 0.4, 0.2, 0.1, 0}
5	{ $O_{52}, O_{56}, O_{54}, O_{51}, O_{53}, O_{57}, O_{55}$ }	{6, 3, 7, 8, 9, 13, 6}	{0, 0.4, 0.6, 0.4, 0.2, 0.3, 0.2}
6	{ $O_{66}, O_{62}, O_{65}, O_{64}, O_{61}, O_{63}, O_{67}$ }	{1, 4, 5, 10, 11, 4, 12}	{0.4, 0, 0.2, 0.6, 0.4, 0.1, 0.3}
7	{ $O_{74}, O_{76}, O_{77}, O_{73}, O_{72}, O_{75}, O_{71}$ }	{11, 12, 15, 8, 16, 7, 13}	{0.6, 0.4, 0.3, 0.1, 0, 0.2, 0.4}

表3 实例3的实验数据  
Table 3 Experimental data of the third instance

$J$	$A_j$	$P_{ji}$	$\alpha_{ji}$
1	$\{O_{18}, O_{17}, O_{16}, O_{11}, O_{13}\}$	$\{7, 20, 17, 4, 8\}$	$\{0.3, 0.3, 0.4, 0.4, 0.1\}$
2	$\{O_{23}, O_{27}, O_{24}, O_{26}, O_{22}, O_{28}\}$	$\{3, 19, 13, 1, 9, 15\}$	$\{0.1, 0.3, 0.6, 0.4, 0, 0.4\}$
3	$\{O_{37}, O_{33}, O_{34}, O_{35}\}$	$\{14, 5, 12, 18\}$	$\{0.3, 0.1, 0.6, 0.2\}$
4	$\{O_{44}, O_{45}, O_{42}, O_{46}, O_{47}, O_{43}, O_{41}, O_{48}\}$	$\{17, 9, 6, 9, 16, 5, 8, 20\}$	$\{0.6, 0.2, 0, 0.4, 0.3, 0.1, 0.4, 0.3\}$
5	$\{O_{51}, O_{56}, O_{55}, O_{53}, O_{58}\}$	$\{10, 7, 20, 17, 14\}$	$\{0.4, 0.4, 0.2, 0.1, 0.3\}$
6	$\{O_{66}, O_{63}, O_{64}, O_{67}, O_{65}, O_{62}, O_{68}, O_{61}\}$	$\{13, 14, 8, 19, 16, 18, 14, 14\}$	$\{0.4, 0, 0.6, 0.3, 0.2, 0, 0.3, 0.4\}$
7	$\{O_{73}, O_{72}, O_{71}, O_{75}, O_{74}, O_{78}, O_{77}, O_{76}\}$	$\{10, 18, 10, 7, 9, 7, 9, 7\}$	$\{0.1, 0, 0.4, 0.2, 0.6, 0.3, 0.3, 0.4\}$
8	$\{O_{81}, O_{85}, O_{88}, O_{86}, O_{82}\}$	$\{13, 1, 15, 9, 4\}$	$\{0.4, 0.2, 0.3, 0.4, 0\}$

### 3.1 回溯策略对 NP 算法的影响

基于标准的 NP 算法, 验证回溯策略 1) 和 2) 及抽样个数对算法的影响. 从表 4 (SN 表示抽样个数) 可以看出, 两种策略均能在可接受的时间内获得相同的最好解. 从均值和标准方差上看, 策略 2) 的搜索效果优于策略 1), 且其稳定性较好. 另外, 增加抽样个数可以提高算法的求解效果和稳定性, 但其求解时间也会相应增加. 为提高算法的稳定性, 本文选取策略 2) 作为后续实验的回溯策略.

### 3.2 抽样个数对 NP-PGA 的影响

为验证抽样个数对 NP-PGA 的影响, 固定 PGA 的迭代次数为 50, 且采取回溯策略 2). 表 5 (NG 为迭代次数) 给出了实验结果. 从表 5 可以看出, 抽样个数对 NP-PGA 影响的趋势类似于第 3.1 节中的结论, 五种抽样个数的设置均能获得相同的最好解, 从均值和标准方差上看, 当抽样个数取 30 时, 算法的稳定性较好. 因此, 在本文的后续实验中将抽样个数设置为 30.

### 3.3 PGA 迭代次数的影响

为验证 PGA 迭代次数对算法的影响, 固定抽样个数为 30, 且采取回溯策略 2). 表 6 为验证 PGA 的迭代次数对 NP-PGA 性能影响的实验结果. 从表 6 中可以看出, 随着迭代次数增大, 算法的求解效果得到改善, 均值和标准方差均有所下降, 但也增大了计算开销, 导致算法的求解时间增大. 另外, 当迭代次数为 30 时, NP-PGA 的稳定性较好. 因此, 在后续实验中将 PGA 的迭代次数设置为 30.

### 3.4 算法对比

为验证 NP-PGA 的有效性, 分别采用枚举算法 (EA)、遗传算法 (GA)、NP 算法及 NP-PGA 求解实例 1~3. 其中, EA 可以获得问题的最优解, 用

以验证 NP-PGA 获得最优解的能力; GA 已被证明可以有效求解作业车间调度问题, 并被众多研究作为对比算法<sup>[15-16, 20]</sup>, NP 算法用于验证本文所提出的改进策略的有效性. GA 的参数设置为: 种群规模为 100, 交叉率为 0.7, 变异率为 0.3, 迭代次数为 500. NP-PGA 采取回溯策略 2), 抽样次数为 30, PGA 迭代次数为 30. NP 的参数设置同 NP-PGA. 表 7 给出了求解的实验结果. 对于所采用的 3 个实例 (“-”表示 EA 不能在有效的时间内获得问题最优解), 将 NP-PGA 得到的结果与 EA、GA 和 NP 得到的结果进行对比. 对于 EA 算法,  $J6 \times M6$  问题计算最优解的时间为 3237.8s, 并且该方法的计算结果与使用 GA、NP 和 NP-PGA 计算得到的结果一致. 对于  $J7 \times M7$  问题, EA 不能在有效的时间内获得最优解, 其他 3 种算法均能够得到近似解. 但是, 显而易见的是, NP-PGA 的求解结果优于 GA 和 NP 的求解结果, 但求解时间相对较大. 对于  $J8 \times M8$  问题, NP-PGA 的求解结果优势更加明显, 从最好解、均值、标准方差及求解时间上看, NP-PGA 均好于 GA 和 NP. 对于 NP 算法, 由于其采用随机抽样的方法, 不能保证获得的样本解为可行域的最好解, 从而造成回溯, 而多次回溯必将会影响到算法的执行效率. 针对这一问题, 在抽样阶段嵌入 PGA 方法对 NP 加以改进. 从表 7 的数据可以看出, NP-PGA 算法与 NP 算法相比, 最好解、均值、标准方差及求解时间均得到改进, 并且随着问题规模的增大, NP-PGA 的优势更加明显. 通过在 NP 算法中嵌入 PGA, 可以提高抽样的多样性及样本质量, 尽可能获得能够代表可行域的最好解, 从而减少了回溯次数, 提高了算法的求解性能.

为直观地展示所提出的 NP-PGA 算法的有效性, 分别选取三种算法 30 次求解的最好结果绘制收敛曲线. 图 5~7 分别给出了三种算法在求解  $J6 \times M6$ ,  $J7 \times M7$  及  $J8 \times M8$  问题的收敛曲线图, 图

表 4 回溯策略及抽样个数对算法影响

Table 4 Experimental results on different backtracking strategies and sampling number

策略	SN	Best	Worst	Mean	S	Time (s)
1)	20	178.83	219.87	192.41	16.03	19.23
	30	178.83	205.91	185.85	9.49	25.51
	40	178.83	204.40	185.01	10.78	29.26
	50	178.83	192.24	183.83	10.12	39.33
	60	178.83	184.96	181.77	8.99	53.80
2)	20	178.83	221.46	190.07	9.06	14.37
	30	178.83	198.10	185.31	9.85	24.43
	40	178.83	215.10	184.73	9.32	38.00
	50	178.83	195.34	182.55	8.96	57.42
	60	178.83	183.51	180.22	8.37	88.57

表 5 抽样个数对算法影响

Table 5 Experimental results on different sampling number

SN	NG	Best	Worst	Mean	S	Time (s)
10	50	178.83	210.69	186.57	12.94	17.74
20	50	178.83	198.09	186.03	12.32	36.50
30	50	178.83	192.10	181.69	5.49	71.13
40	50	178.83	192.10	182.80	7.31	85.09
50	50	178.83	192.10	181.43	5.74	99.76

表 6 PGA 迭代次数对算法影响

Table 6 Experimental results on different iteration times of PGA

SN	NG	Best	Worst	Mean	S	Time (s)
30	10	178.83	210.69	189.43	13.25	22.51
30	20	178.83	210.69	186.99	10.13	30.64
30	30	178.83	188.84	183.21	4.97	45.19
30	40	178.83	192.10	183.83	5.81	65.59
30	50	178.83	198.09	184.99	7.71	82.45

表 7 EA、GA、NP 和 NP-PGA 算法对比结果

Table 7 Comparison of experimental results obtained by EA, GA, NP and NP-PGA

Problem	Algorithm	Best	Mean	S	Time (s)
$J6 \times M6$	EA	178.83	178.83	0	3 237.8
$J6 \times M6$	GA	178.83	181.38	1.92	33.39
$J6 \times M6$	NP	178.83	192.10	9.06	38.03
$J6 \times M6$	NP-PGA	178.83	190.07	5.49	45.19
$J7 \times M7$	EA	—	—	—	—
$J7 \times M7$	GA	894.80	917.08	19.50	51.68
$J7 \times M7$	NP	901.51	921.75	11.82	65.77
$J7 \times M7$	NP-PGA	891.30	910.08	11.20	74.52
$J8 \times M8$	EA	—	—	—	—
$J8 \times M8$	GA	838.60	842.76	10.22	95.88
$J8 \times M8$	NP	833.43	873.42	17.26	119.42
$J8 \times M8$	NP-PGA	805.28	820.86	10.04	88.93

中横坐标表示算法运行时间, 纵坐标表示算法搜索到的最好解. 从图中可以看出, NP-PGA 收敛的速度和效果均要优于 GA 和 NP, 且在求解  $J8 \times M8$  问题上优势更加明显.

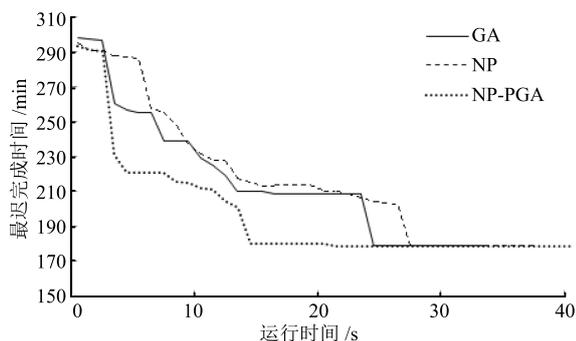


图5 三算法求解  $J6 \times M6$  收敛曲线

Fig. 5 Convergence curve of three algorithms in solving  $J6 \times M6$

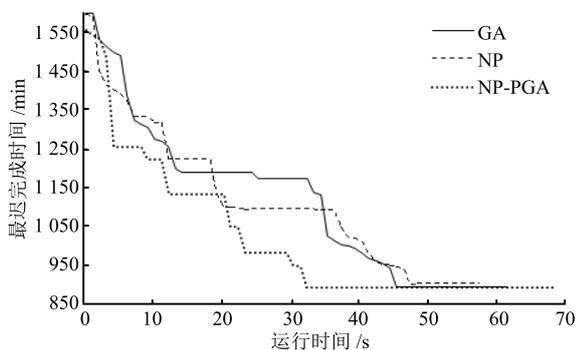


图6 三算法求解  $J7 \times M7$  收敛曲线

Fig. 6 Convergence curve of three algorithms in solving  $J7 \times M7$

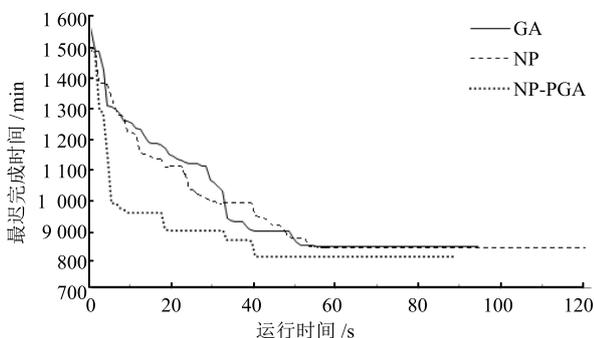


图7 三算法求解  $J8 \times M8$  收敛曲线

Fig. 7 Convergence curve of three algorithms in solving  $J8 \times M8$

## 4 结论

设备带有恶化特性的作业车间调度问题是制造企业亟需解决的关键问题. 本文针对设备带有恶

化特性的作业车间调度问题进行了研究. 首先, 对作业车间环境下设备带有恶化特性的调度问题进行了描述, 建立了以最小化最迟完成时间为目标的问题模型; 进而针对问题特点设计了嵌入单亲遗传算法的混合嵌套分割算法进行问题求解; 最后, 通过与枚举算法、遗传算法及标准嵌套分割算法的对比分析, 表明所提出的算法在求解质量、求解时间和稳定性方面均具有较好的性能, 尤其在作业和设备数量增多时, 所提出的算法具有更加明显的优势. 该研究为设备带有恶化特性的作业车间调度问题提供了有效的建模和求解工具, 在自动和手控设备并存的作业车间调度方面具有广泛的应用. 未来研究可进一步分析设备恶化系数对算法性能的影响等.

## References

- 1 Brucker P. *Scheduling Algorithms*. Berlin: Springer-Verlag, 2007. 69–83
- 2 Blazewicz J, Domschke W, Pesch E. The job shop scheduling problem: conventional and new solution techniques. *European Journal of Operational Research*, 1996, **93**(1): 1–33
- 3 Wang L, Tang D B. An improved adaptive genetic algorithm based on hormone modulation mechanism for job-shop scheduling problem. *Expert Systems with Applications*, 2011, **38**(6): 7243–7250
- 4 Qiao Wei, Wang Bing, Sun Jie. Uncertain job shop scheduling problems solved by genetic algorithm. *Computer Integrated Manufacturing Systems*, 2007, **13**(12): 2452–2455 (乔威, 王冰, 孙洁. 用遗传算法求解一类不确定性作业车间调度问题. 计算机集成制造系统, 2007, **13**(12): 2452–2455)
- 5 Li Fu-Ming, Zhu Yun-Long, Yin Chao-Wan, Song Xiao-Yu. Research on fuzzy job shop scheduling with alternative machines. *Computer Integrated Manufacturing Systems*, 2006, **12**(2): 169–173 (李富明, 朱云龙, 尹朝万, 宋晓宇. 可变机器约束的模糊作业车间调度问题研究. 计算机集成制造系统, 2006, **12**(2): 169–173)
- 6 Yan Li-Jun, Li Zong-Bin, Wei Jun-Hu, Du Xuan. A new hybrid optimization algorithm and its application in job shop scheduling. *Acta Automatica Sinica*, 2008, **34**(5): 604–608 (闫利军, 李宗斌, 卫军胡, 杜轩. 一种新的混合优化算法及其在车间调度中的应用. 自动化学报, 2008, **34**(5): 604–608)
- 7 Gupta J N D, Gupta S K. Single facility scheduling with nonlinear processing times. *Computers and Industrial Engineering*, 1988, **14**(4): 387–393
- 8 Wu H P, Huang M. Improved estimation of distribution algorithm for the problem of single-machine scheduling with deteriorating jobs and different due dates. *Computational and Applied Mathematics*, 2014, **33**(3): 557–573
- 9 Mosheiov G. Scheduling jobs under simple linear deterioration. *Computers and Operations Research*, 1994, **21**(6): 653–659
- 10 Wu C C, Wu W H, Wu W H, Hsu P H, Yin Y Q, Xu J Y. A single-machine scheduling with a truncated linear deterioration and ready times. *Information Sciences*, 2014, **256**: 109–125

- 11 Cheng M B, Tadikamalla P R, Shang J, Zhang S Q. Bi-criteria hierarchical optimization of two-machine flow shop scheduling problem with time-dependent deteriorating jobs. *European Journal of Operational Research*, 2014, **234**(3): 650–657
- 12 Wang J B, Wang M Z. Solution algorithms for the total weighted completion time minimization flow shop scheduling with decreasing linear deterioration. *The International Journal of Advanced Manufacturing Technology*, 2013, **67**(1–4): 243–253
- 13 Liu C H, Chen L S, Lin P S. Lot streaming multiple jobs with values exponentially deteriorating over time in a job-shop environment. *International Journal of Production Research*, 2013, **51**(1): 202–214
- 14 Mosheiov G. Complexity analysis of job-shop scheduling with deteriorating jobs. *Discrete Applied Mathematics*, 2002, **117**(1–3): 195–209
- 15 Liu C H. Scheduling jobs with values exponentially deteriorating over time in a job shop environment. In: Proceedings of the 2011 International MultiConference of Engineers and Computer Scientists. Hong Kong, China: Newswood Limited, 2011. 1113–1118
- 16 Araghi M E T, Jolai F, Rabiee M. Incorporating learning effect and deterioration for solving a SDST flexible job-shop scheduling problem with a hybrid meta-heuristic approach. *International Journal of Computer Integrated Manufacturing*, 2013, **27**(8): 733–746
- 17 Garey M R, Johnson D S, Sethi R. The complexity of flow shop and job shop scheduling. *Mathematics of Operations Research*, 1976, **1**(2): 117–129
- 18 Shi L, Ólafsson S. *Nested Partitions Method, Theory and Applications*. New York: Springer-Verlag, 2008. 131–226
- 19 Shi L, Ólafsson S. Nested partitions method for global optimization. *Operations Research*, 2000, **48**(3): 390–407
- 20 Wang Y M, Yin H L, Qin K D. A novel genetic algorithm for flexible job shop scheduling problems with machine disruptions. *The International Journal of Advanced Manufacturing Technology*, 2013, **68**(5–8): 1317–1326



**黄敏** 东北大学信息科学与工程学院教授。主要研究方向为生产计划、调度与存储控制, 物流与供应链管理, 行为运筹, 风险管理和软计算。

E-mail: mhuang@mail.neu.edu.cn  
**(HUANG Min)** Professor at the College of Information and Engineering, Northeastern University. Her

research interest covers production planning, scheduling and

inventory control, behavioral operation, management of logistics and supply chain, and risk management and soft computing.)

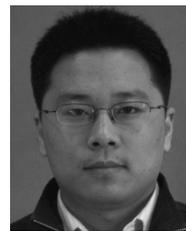


**付亚平** 东北大学信息科学与工程学院系统工程研究所博士研究生。主要研究方向为生产计划与调度, 智能优化算法。本文通信作者。

E-mail: fuyaping0432@163.com

**(FU Ya-Ping)** Ph.D. candidate at the College of Information and Engineering, Northeastern University. His

research interest covers production planning and scheduling, optimization algorithm. Corresponding author of this paper.)



**王洪峰** 东北大学信息科学与工程学院副教授。主要研究方向为进化计算, 生产计划与调度, 物流与供应链管理。

E-mail: hfwang@mail.neu.edu.cn

**(WANG Hong-Feng)** Associate professor at the College of Information and Engineering, Northeastern University. His research interest covers evolu-

tionary algorithm, production planning and scheduling, and management of logistics and supply chain.)



**朱兵虎** 东北大学信息科学与工程学院系统工程研究所硕士研究生。主要研究方向为生产计划与调度, 智能优化算法。

E-mail: zhubinghu0311@163.com

**(ZHU Bing-Hu)** Master student at the College of Information and Engineering, Northeastern University. His research interest covers production

planning and scheduling, optimization algorithm.)



**王兴伟** 东北大学信息科学与工程学院教授。主要研究方向为下一代互联网, 光互联网, 移动互联网。

E-mail: wangxw@mail.neu.edu.cn

**(WANG Xing-Wei)** Professor at the College of Information and Engineering, Northeastern University. His research interest covers next generation

internet (NGI), optical internet, and mobile internet.)