

改进算术优化算法求解考虑机器老化效应和 工件释放时间的作业车间调度问题*

陈照辉¹, 刘海杨², 夏 倩³, 张新功⁴

(1. 重庆科技大学 数理与大数据学院; 2. 重庆科技大学 智能技术与工程学院, 重庆 401331;
3. 重庆谢家湾学校, 重庆 400050; 4. 重庆师范大学 数学科学学院, 重庆 401331)

摘要: 研究考虑机器具有老化效应和工件带有释放时间约束的作业车间调度问题(job shop-scheduling problem, JSP)。建立以最小化最大完工时间为目标的调度优化模型, 并设计 1 种改进的算术优化算法(improved arithmetic optimization algorithm, IAOA)对该问题进行求解。算法首先通过排序值转换规则将 IAOA 连续解空间映射到 JSP 的离散空间, 并对 JSP 进行编码, 然后使用插入式贪婪解码算法进行解码。提出了非线性数学优化加速函数和 6 种邻域搜索策略对标准算术优化算法(arithmetic optimization algorithm, AOA)进行改进。通过在 33 个 JSP 数据集上进行测试并与 AOA、灰狼优化算法和算术三角函数优化算法进行对比分析, 结果表明提出的 IAOA 具有较好的优化效果以及收敛能力, 且该算法克服了 AOA 求解精度低、收敛速度慢的缺陷。

关键词: 算术优化算法; 作业车间调度问题; 插入式贪婪解码算法; 机器老化效应; 释放时间

中图分类号: O224; TP391

文献标志码: A

文章编号: 1672-6693(2024)02-0056-09

车间调度问题作为如半导体制造过程^[1]、汽车装配过程^[2]、机械制造系统^[3]等现代生产环境的高度抽象, 在过去的几十年中得到了广泛的研究^[4-5]。带有释放时间约束的作业车间调度问题(job shop-scheduling problem, JSP)是车间调度问题的一种, 目标是在满足问题约束的同时, 寻求一种理想的加工调度方案, 使得全部工件加工完成时间最短。JSP 引起了学者们极大的兴趣, 不仅是因为它可以作为一个框架来推导制造系统的有效调度策略, 还因为它可以作为一个具有挑战性的基准用于 NP-困难组合优化问题^[6]。

目前解决作业车间调度问题的算法主要分为精确式算法、启发式算法和元启发式算法。精确式算法包括分支定界法、动态规划等方法^[7], 对于小规模数据问题, 精确式算法可以得到理论最优解。但是由于算法的时间和空间复杂度较高, 往往难以应用在实际问题当中。启发式算法是通过过去经验的归纳推理以及实验分析来解决问题, 即借助于某种直观判断或试探, 以求得问题的次优解, 例如加工时间最短的工件优先加工(shortest processing time, SPT)规则^[8]就属于该类算法。虽然启发式算法可以很快求得问题的解, 但是很难保证算法的收敛性。元启发式算法由于原理简单, 对初始值不敏感且易于实现, 因此被广泛地应用于处理非线性优化问题。此外, 元启发式算法不依赖于目标函数的梯度, 这也使得解决方案的适用性更有说服力。

Tasgetiren 等人^[9]提出使用粒子群算法以最小化最大完成时间为目标来解决 JSP。Zhang 等人^[10]提出一种人工蜂群算法求解 JSP, 目标是在工件具有交货日期约束下的最小化总加权误工损失。薛玲玲^[11]提出基于块结构邻域搜索的遗传算法去解决 JSP, 并增加了个体冗余性的判断条件, 使得算法性能得到了极大的改善。王艳红等人^[12]利用决策树算法在生产数据中提取特征, 并运用在作业车间调度中, 再通过经典的 JSP 测试数据集验证了算法的可行性。

目前所研究的对象更多的是针对于经典的车间调度问题。但是在实际的制造业工厂中, 面临的情况更加复杂。以某著名 PCB 电路板加工工厂为例, 影响生产效率主要有 3 个方面: 1) 工件的加工顺序; 2) 工件的释放时间; 3) 机器的加工速率。其中, 工件的释放时间是指原材料到达加工工厂的时间。而机器的加工速率并不是固定的, 因

* 收稿日期: 2022-10-28 修回日期: 2023-12-27 网络出版时间: 2023-09-21T14:51

资助项目: 国家自然科学基金——重大项目(No. 11991022), 面上项目(No. 11971443); 重庆市自然科学基金(No. cstc2021jcyj-msxmX0229); 重庆市教育委员会科学技术研究计划——重点项目(No. KJZD-K202000501, No. KJZD-K202301502), 青年项目(No. KJQN202001507)

第一作者简介: 陈照辉, 男, 副教授, 博士, 研究方向为智能优化算法、智能控制, E-mail: z.chen@cqust.edu.cn; 通信作者: 张新功, 男, 教授, 博士, E-mail: zhangxingong@cquu.edu.cn

网络出版地址: <https://link.cnki.net/urlid/50.1165.N.20230920.1900.006>

为随着机器使用次数的增加,会出现老化的情况,从而导致机器的加工速率降低,这种情况称为机器的老化效应。

为了更加切合实际企业的需求,本文研究了以最小化最大完工时间为目标,考虑机器老化效应和工件释放时间的作业车间调度问题,并在标准算术优化算法(arithmetic optimization algorithm, AOA)基础上提出一种改进的算术优化算法(improved arithmetic optimization algorithm, IAOA)解决该问题。

AOA 是一种基于种群模型的元启发式算法^[13],搜索过程分为探索阶段和开发阶段。在探索阶段,利用算术运算中的乘法和除法运算符的分布特性扩大全局搜索的分散性;在开发阶段,则通过加法和减法运算提高局部搜索的准确性。近些年来,AOA 已经被大量的运用到求解实际问题当中,例如控制器设计^[14]、电力系统^[15]和数据聚类^[16]等。但 AOA 仍存在一些不足,例如求解精度低、收敛速度慢等。有鉴于此,本文提出采用 6 种邻域搜索结构和非线性数学优化加速(mathematical optimization acceleration, MOA)函数进一步提高 AOA 的收敛能力和泛化能力。

1 问题描述及数学模型

1.1 问题描述

本文考虑工件带释放时间和机器具有老化效应的 JSP,该调度问题可以描述 n 个工件在 m 台机器上进行加工。每个工件均有 m 道工序 $O_i = \{o_{i1}, o_{i2}, \dots, o_{im}\} (i=1, 2, \dots, n)$, 并且工序的顺序是已知的。工序 $o_{ij} (i=1, 2, \dots, n; j=1, 2, \dots, m)$ 在指定的机器上加工,并且在该工件的加工完成之前不能被任何其它作业中断。工件 i 的释放时间为 $r_i (i=1, 2, \dots, n)$, 如果某工件没有释放,则当前工件不能进行加工。每台机器均存在老化效应,老化效应是指每台机器随着使用次数的增多而导致工件的加工时长延长。每个工件的不同工序完工时间被定义为 $c = \{c_{i1}, c_{i2}, \dots, c_{im}\}$, 则工件 i 的最大完工时间为 c_{im} , 所有工件全部完成的最大完工时间为 $\max_i \{c_{im}\}$ 。

1.2 模型建立

本文以最小化最大完工时间作为优化目标^[17],同时考虑工件的释放时间和机器的老化效应。记

$$x_{ijk} = \begin{cases} 1, & \text{如果第 } i \text{ 个工件的第 } j \text{ 个工序在指定的机器 } k \text{ 上加工;} \\ 0, & \text{否则} \end{cases};$$

$$y_{ijhkl} = \begin{cases} 1, & \text{如果 } o_{ij} \text{ 先于 } o_{hl} \text{ 在指定的机器 } k \text{ 上加工} \\ 0, & \text{否则} \end{cases} \quad (i \neq h)。$$

其中: p_{ij} 表示第 i 个工件的第 j 个工序的加工时间, c_{ij} 表示第 i 个工件的第 j 道工序完工时间, s_i 表示第 i 个工件的开始时间, q_{ij} 表示第 i 个工件的第 j 个工序的开始时间, β 为机器的老化因子, u_{ik} 表示第 i 个工件在机器 k 上的加工位置。建立如下数学模型:

$$\min c_{\max} = \max_i \{c_{im}\}, \quad (1)$$

$$\text{s. t. } p_{ij} \geq 0, \quad (2)$$

$$s_i \geq r_i, \quad (3)$$

$$q_{ij} + (1 + (u_{ik})^\beta) p_{ij} x_{ijk} \leq c_{ij}, \quad (4)$$

$$q_{i(j+1)} - c_{ij} \geq 0, \quad (5)$$

$$c_{ij} \leq c_{\max}, \quad (6)$$

$$c_{hl} - c_{ij} + L(1 - y_{ijhkl}) \geq p_{hl}, \quad (7)$$

其中: $i, h=1, 2, \dots, n; j, l, k=1, 2, \dots, m$; 式(1)中 $c_{\max}$ 为目标函数, 目标为最小化最大完成时间; 式(2)表示每道工序的加工时间大于等于 0; 式(3)表示工件在释放时间之后才能加工; 式(4)中考虑机器 k 的老化效应, 它和式(5)都表示每个工件的工序存在先后顺序; 式(6)表示每一工件的完工时间不能超过总的完工时间; 式(7)中 L 是一个足够大的整数, 该式表示不同工件在同一机器上具有先后顺序。

2 IAOA

2.1 基本 AOA

2.1.1 初始化

AOA 的初始种群记为 $X_1, X_2, \dots, X_N$, 其中: 第 i 个个体为 $X_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$, N 表示种群规模, D 表示每一个个体的维度。

AOA 通过数学加速函数 MOA 函数选择下一步搜索策略。MOA 函数记为 $A_{\text{MOA}}(c)$, 计算公式如下:

$$A_{\text{MOA}}(c) = A_{\min} + c * \left(\frac{A_{\max} - A_{\min}}{T} \right), \quad (8)$$

其中: c 表示当前迭代次数, T 表示迭代总次数, $A_{\text{MOA}}(c)$ 表示第 c 代的加速函数, $A_{\min}$ 和 $A_{\max}$ 分别为加速函数的最小值和最大值, 取值分别为 0.2 和 1。

2.1.2 探索与开发

在 AOA 中, 如果随机数 $r_1 < A_{\text{MOA}}(c)$, 则 AOA 通过乘法和除法运算策略探索整个搜索空间以找到更好的候选解。如果随机数 $r_2 < 0.5$, 则执行除法运算。否则, 如果随机数 $r_2 \geq 0.5$ 时, 将执行乘法运算, 位置更新方式如式(9)所示。否则, 如果 $r_1 \geq A_{\text{MOA}}(c)$, AOA 将通过加法和减法操作探索局部空间, 以找到更好的候选解。如果随机数 $r_3 < 0.5$, 则将执行减法运算。否则, 如果随机数 $r_3 \geq 0.5$ 时, 将执行加法运算, 位置更新方式如式(10)所示。

$$x_{i,j}(c+1) = \begin{cases} f_{\text{best}}(x_j) \div (\sigma_{\text{MOP}}(c) + \epsilon) \times ((U_j - L_j) \times \mu + L_j), & r_2 < 0.5 \\ f_{\text{best}}(x_j) \times \sigma_{\text{MOP}}(c) \times ((U_j - L_j) \times \mu + L_j), & r_2 \geq 0.5 \end{cases}; \quad (9)$$

$$x_{i,j}(c+1) = \begin{cases} f_{\text{best}}(x_j) - \sigma_{\text{MOP}}(c) \times ((U_j - L_j) \times \mu + L_j), & r_3 < 0.5 \\ f_{\text{best}}(x_j) + \sigma_{\text{MOP}}(c) \times ((U_j - L_j) \times \mu + L_j), & r_3 \geq 0.5 \end{cases} \quad (10)$$

其中: $x_{i,j}(c+1)$ 表示在第 $c+1$ 次迭代时第 i 个解的第 j 个位置, $f_{\text{best}}(x_j)$ 表示到当前迭代次数为止获得的最优解的第 j 个位置, ϵ 是一个小的整数, U_j 和 L_j 分别表示最优解在第 j 维的上界和下界, μ 是一个控制参数, 值为 0.449, $\sigma_{\text{MOP}}(c)$ 代表数学优化概率系数, 表达式为

$$\sigma_{\text{MOP}}(c) = 1 - \frac{c^{1/a}}{T^{1/a}}, \quad (11)$$

其中: $\sigma_{\text{MOP}}(c)$ 表示第 c 次迭代时的函数值。参数 α 是一个敏感参数, 值设置为 5。

2.2 算法的改进

2.2.1 基于非线性 MOA 函数收敛策略

由于 AOA 的 MOA 函数从 0.2 线性递增到 1, 但是算法在不断收敛的过程中并不是线性的, MOA 函数线性递增策略不能完全体现出实际的优化搜索过程。基于此, 本文设计了一种新的非线性 MOA 函数, 表达式为:

$$A'_{\text{MOA}}(c) = A_{\min} + (A_{\max} - A_{\min}) \times \left(\frac{1}{e-1} \times (e^{c/T} - 1) \right), \quad (12)$$

本文构造的改进非线性 MOA 函数与原 MOA 函数对比如图 1 所示。

2.2.2 邻域搜索策略

为了进一步增强 AOA 用于解决作业车间调度问题的搜索能力, 本文对工序编码 $X_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$ 使用以下 6 种策略进行邻域搜索。

1) 邻域搜索结构 N_1 (两点交叉): 在工序编码上任选 2 个对应不同工序的元素, 然后对它们进行交换操作。

2) 邻域搜索结构 N_2 (前插): 在工序编码中任选 2 个元素, 然后将后一元素插入到前一元素之前的位置。

3) 邻域搜索结构 N_3 (后插): 在工序编码中任选 2 个元素, 然后将前一元素插入到后一元素之前的位置。

4) 邻域搜索结构 N_4 (前后工序块随机交换): 选择 1 个随机数 $\gamma_1 \in \left[0, \frac{D}{2}\right]$, 将 $[x_{i1}, x_{i2}, \dots, x_{i\gamma_1}]$ 和 $[x_{i(n-\gamma_1)}, x_{i(n-\gamma_1+1)}, \dots, x_{iD}]$ 这 2 个工序块进行互换。如图 2 所示。

5) 邻域搜索结构 N_5 (随机工序块互换): 设置随机整数 $\gamma_2 \in \left[1, \frac{D}{2}\right]$, $\gamma_3 \in \left[\frac{D}{2}+1, D\right]$, $\gamma_4 \in \left[1, \min \left\{ \frac{D}{2} - \gamma_2, D - \gamma_3 \right\} \right]$, 交换序列 $[x_{i\gamma_2}, x_{i(\gamma_2+1)}, \dots, x_{i\gamma_4}]$ 和 $[x_{i\gamma_3}, x_{i(\gamma_3+1)}, \dots, x_{i\gamma_4}]$ 。

6) 邻域搜索结构 N_6 : 删除最优解和最差解的相同部分, 并用当前解进行填充, 如图 3 所示。

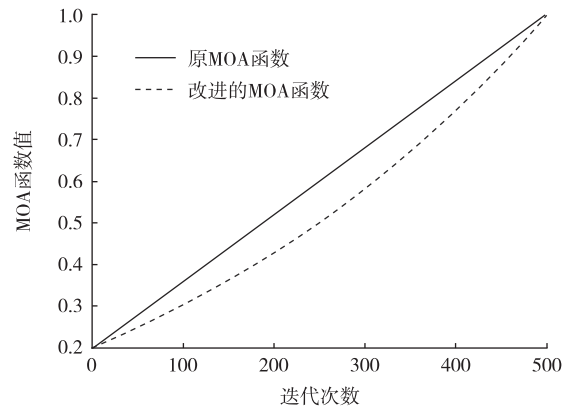


图 1 MOA 函数对比

Fig. 1 Comparison chart of MOA functions

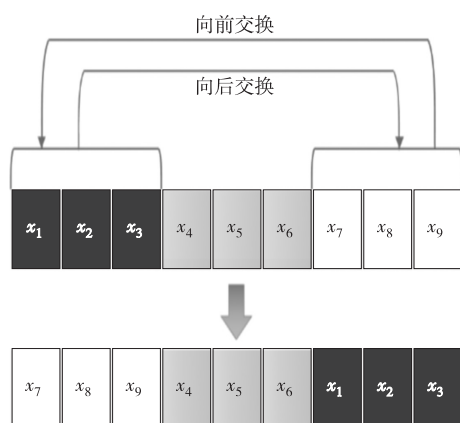
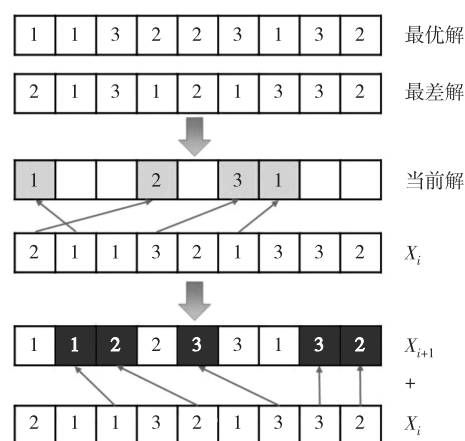


图 2 前后工序块随机交换

Fig. 2 Random swap of preceding and following process blocks

图 3 邻域搜索结构 N_6 Fig. 3 Neighborhood search structure N_6

2.3 算法流程

IAOA 的伪代码如表 1 所示。

表 1 IAOA

Tab. 1 Improved arithmetic optimization algorithm

行号	伪代码
1	设定 IAOA 参数 α, μ
2	初始化种群
3	While ($C < T$)
4	计算每个个体的适应度值
5	寻找最优解
6	使用式(12)更新 $MOA(c)$ 值
7	使用式(11)更新 $MOP(c)$ 值
8	For ($i = 1: N$)
9	For ($j = 1: D$)
10	If $r_1 > A_{MOA}(c)$
11	If $r_2 < 0.5$
12	执行式(9)中的除法搜索策略
13	Else
14	执行式(9)中的乘法搜索策略
15	End
16	Else
17	If $r_3 < 0.5$
18	执行式(10)中的减法搜索策略
19	Else
20	执行式(10)中的加法搜索策略
21	End
22	End
23	End
24	End
25	执行 6 种邻域搜索策略
26	$C = C + 1$
27	End

3 IAOA 在 JSP 中的应用

3.1 编码

JSP 是一个离散优化问题,IAOA 主要用于连续空间问题的求解。因此,如果想通过 IAOA 去解决 JSP 调度问题,就需要建立连续空间与离散空间的映射关系。本文采用 ROV 转换^[18]将连续解空间映射到 JSP 的离散空间。具体规则如下:首先生成一组连续向量 $\mathbf{X}_i=[x_{i1},x_{i2},\cdots,x_{iD}]$ 和工件编号向量 $\mathbf{J}=[j_1,j_2,\cdots,j_D]$ 。然后挑选出连续向量中最小的位置,并分配工序编号为 1,再次挑选出连续向量中第 2 小的位置,分配工序编号为 2。类似地,将所有的连续向量转换为工序编号。例如,考虑具有 3 台机器 2 个工件的作业车间调度问题。如表 2 所示,连续向量 $\mathbf{X}_i=[0.29,2.31,5.32,3.73,1.13,0.62]$,对应的工序编号为 $[O_1,O_2]=[o_{11},o_{12},o_{13},o_{23},o_{22},o_{21}]$ 。

表 2 ROV 转换
Tab.2 ROV conversion

变量名称	位置向量					
	0.29	2.31	5.32	3.73	1.13	0.62
工件序号	j_1	j_1	j_1	j_2	j_2	j_2
ROV 规则	1	4	6	5	3	2
工序序号	o_{11}	o_{12}	o_{13}	o_{23}	o_{22}	o_{21}

3.2 解码

工序解码采用 Salido^[19]提出的插入式贪婪解码算法。算法步骤如下:首先计算工序的开工时间,每道工序的开工时间等于工件前道工序和机器前道工序完工时间的最大值;然后,查找此工序前面的可用机器空闲时间,在不推迟其他已安排工序开工时间的条件下,将它插入对应机器上最早可行的加工时刻。图 4 为插入式贪婪解码算法调度结果,不难发现此算法可以极大缩短工件的最大完工时间。

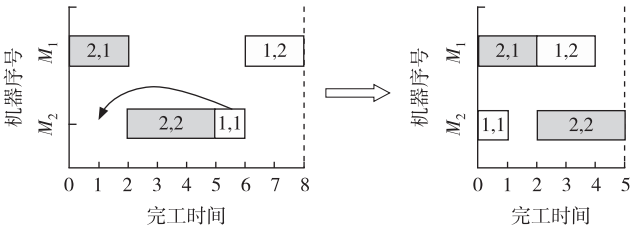


图 4 插入式贪婪解码算法
Fig.2 Insertion-based greedy decoding algorithm

3.3 实验结果

为证明 IAOA 在 JSP 问题中的有效性,本文使用 IAOA 与灰狼优化(grey wolf optimization,GWO)、标准 AOA 和算术三角函数优化算法(arithmetic-trigonometric optimization algorithm,ATOA)^[20]在 33 个测试集上进行比较。其中,种群大小为 30,迭代次数 200 次,每种算法执行 10 次。评价指标为最小值($F_{\min}$)和平均值(F_{mean})。每个工件的释放时间为随机取得 1 个整数,此整数所属区间为 $[0,n]$, n 代表测试集的工件总个数。表 3 列出了每种算法的参数。

表 3 算法参数
Tab.3 Parameters of algorithm

算法	参数
IAOA	$A_{\max}=1,A_{\min}=0.2,\alpha=5,\mu=0.449,\beta=0.01$
AOA	$A_{\max}=1,A_{\min}=0.2,\alpha=5,\mu=0.449$
GWO	$A_{\max}=2,A_{\min}=0$
ATOA	$A_{\max}=1,A_{\min}=0.2,\alpha=5,\mu=0.449$

实验结果如表4所示,相比较于AOA、GWO算法和ATOA,可以发现IAOA的最小化最大完成时间的最优值和平均值在33个测试集上表现的结果均为最优。其中,最优值说明算法的收敛能力,平均值表示算法的整体收敛情况。尤其是针对于多工件多工序调度问题,IAOA表现的更好。例如la26~la30这5个测试集均为20个工件在10台机器上加工,IAOA在 F_{mean} 和 F_{min} 这2个指标上的最大完工时间均提升100 s左右,这也表明了IAOA的优越性和收敛能力。

表4 实验结果
Tab. 4 Results of experimental

算例	IAOA		AOA		GWO		ATOA	
	F_{mean}	F_{min}	F_{mean}	F_{min}	F_{mean}	F_{min}	F_{mean}	F_{min}
FT06	59.027 15	58.795 84	61.267 66	60.551 24	61.9054 8	60.143 33	60.565 29	59.643 25
FT10	1 064.807	1 037.72	1 152.796	1 098.652	1 165.283	1 133.21	1 136.402	1 086.7
FT20	1 394.292	1 357.523	1 546.64	1 457.191	1 546.141	1 504.057	1 504.727	1 466.206
la01	729.959	717.405 3	771.921 8	757.461 6	763.396 2	744.903 1	766.269 9	753.450 1
la02	727.548 4	702.749	792.870 2	756.739 1	791.007 9	771.861 9	780.747 6	724.274 2
la03	680.138 9	665.661 9	723.661 7	718.754 6	701.184 6	684.261 1	708.666 8	689.912 6
la04	648.721 8	632.621 9	680.943 1	661.270 4	693.822 2	664.561 7	676.481 9	663.950 3
la05	620.833	608.725 5	651.366 4	641.964 9	647.845 8	638.179	640.166 1	624.537 5
la06	992.843 4	984.188 1	1 062.451	1 047.101	1 048.112	1 024.693	1 044.95	1 015.722
la07	996.093 4	968.593 4	1 056.703	1 039.724	1 048.463	1 030.441	1 050.363	1 033.103
la08	943.605	920.210 2	999.566 4	989.096 3	993.973 2	960.076	997.387 4	982.841 5
la09	1 054.249	1 030.43	1 110.597	1 082.816	1 098.424	1 080.198	1 099.41	1 089.785
la10	1 027.888	1 012.152	1 087.568	1 072.954	1 081.337	1 067.306	1 077.843	1 056.847
la11	1 330.512	1 307.379	1 424.125	1 395.689	1 403.59	1 389.062	1 416.441	1 391.839
la12	1 173.556	1 150.98	1 232.188	1 210.728	1 220.354	1 193.637	1 225.327	1 214.551
la13	1 273.091	1 254.753	1 335.754	1 313.469	1 324.363	1 293.219	1 325.927	1 301.024
la14	1 367.322	1 349.355	1 460.702	1 439.425	1 450.175	1 410.745	1 445.219	1 408.014
la15	1 377.495	1 352.203	1 484.786	1 462.46	1 470.372	1 441.554	1 456.594	1 395.011
la16	1 072.933	1 055.905	1 145.501	1 106.106	1 126.589	1 103.542	1 125.87	1 082.223
la17	852.205 5	829.160 3	926.685 9	907.163	910.042 9	889.130 6	914.279 8	883.791 3
la18	965.893 2	950.374 2	1 035.643	1 005.382	1017.88 5	995.521 4	1 020.774	999.847 1
la19	946.492 8	922.088 7	1 026.486	1 001.484	1 030.188	1 018.051	1 024.22	1 001.189
la20	1 008.391	970.021 2	1 064.102	1 046.258	1 069.673	1 054.392	1 061.48	1 049.291
la21	1 282.154	1 250.81	1 395.601	1 364.062	1 376.87	1 349.361	1 379.707	1 361.655
la22	1 136.467	1 105.115	1 242.869	1 220.364	1 222.377	1 172.188	1 217.859	1 191.025
la23	1 224.209	1 200.377	1 310.305	1 287.844	1 289.619	1 252.418	1 303.432	1 274.626
la24	1 162.893	1 112.525	1 246.832	1 224.754	1 213.317	1 179.175	1 232.355	1 202.162
la25	1 189.665	1 162.617	1 273.081	1 239.112	1 313.805	1 313.805	1 265.05	1 217.773
la26	1 506.298	1 461.932	1 606.492	1 557.226	1 611.122	1 569.835	1 605.488	1 567.964
la27	1 552.242	1 514.357	1 656.844	1 621.544	1 654.096	1 613.277	1 652.463	1 615.637
la28	1 540.116	1 521.525	1 651.212	1 591.77	1 656.65	1 614.344	1 656.726	1 637.227
la29	1 474.59	1 440.224	1 592.696	1 575.615	1 607.077	1 563.754	1 579.595	1 545.668
la30	1 634.022	1 589.704	1 763.636	1 739.7	1 752.331	1 713.883	1 751.573	1 725.973

为了进一步验证IAOA的寻优性能,实验选取4个测试案例FT06、FT10、la16和la26,画出迭代曲线,如图

5 所示。根据结果可以发现,IAOA 在前期收敛的速度更快并且不容易陷入局部最优解。同时,根据 2 个算例的甘特图(图 6、图 7)可见,IAOA 所得结果充分利用了有限的资源,提高了效率,缩短了加工时长。这也表明本文提出的非线性 MOA 函数策略和 6 种邻域搜索算子是有效的,使得整个算法具有更好的鲁棒性。

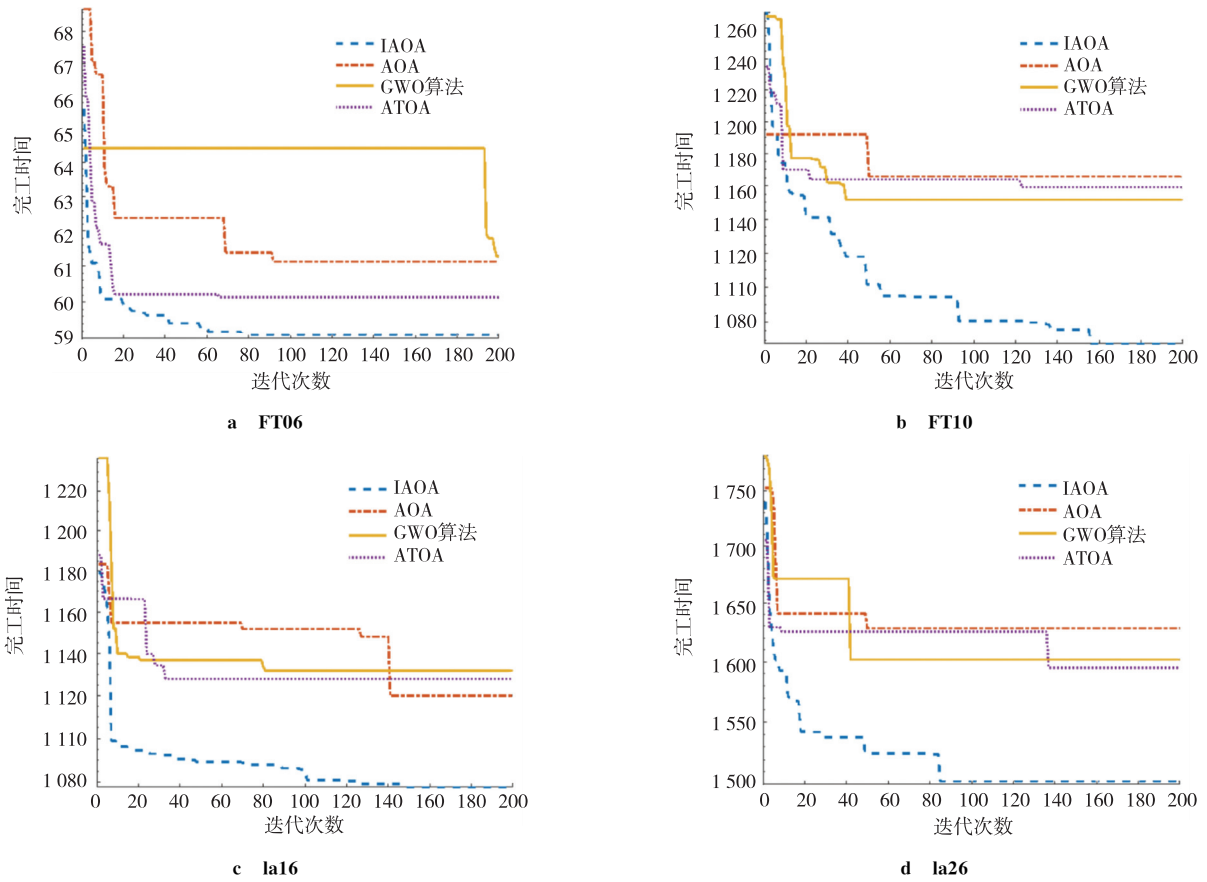


图 5 4 种算法迭代收敛比较

Fig. 5 Iterative convergence comparison of four algorithms

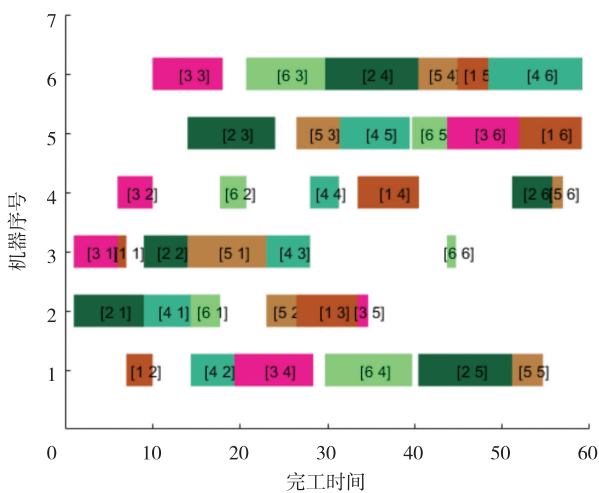


图 6 FT06 甘特图

Fig. 6 Gantt chart of FT06

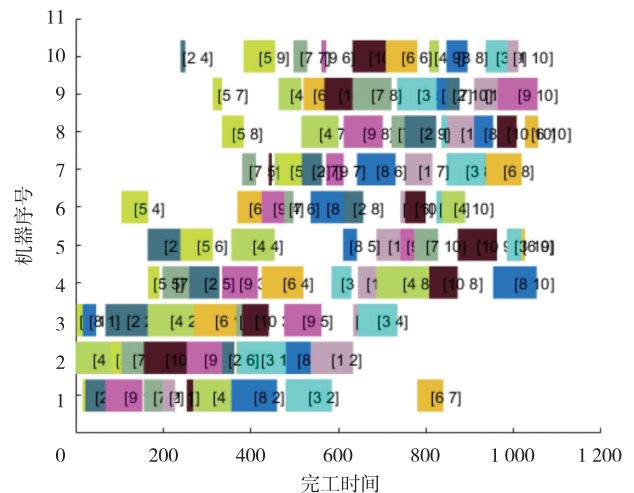


图 7 FT10 甘特图

Fig. 7 Gantt chart of FT10

4 总结

本文提出一种 IAOA,针对带有机老化效应和释放时间的作业车间调度问题,以最小化最大完工时间为目

标,建立调度优化模型,并进行求解。考虑 JSP 问题特点,基于 ROV 转换规则,实现了连续空间和离散空间的映射,使得 IA OA 可以用于求解 JSP 问题。另外,设计 6 种邻域搜索结构和非线性 MOA 函数策略,使得算法的性能得到进一步增强。根据 33 个数据集测试结果表明 IA OA 具有较强的收敛能力和解决 JSP 问题的有效性。研究为连续优化算法在求解考虑机器老化效应和释放时间的作业车间调度问题提供了一种新的解决思路。

参考文献:

- [1] 周炳海,周琪,郑雯. 半导体制造 AMHS 中悬挂式小车多因素调度方法[J]. 北京理工大学学报,2014,34(6):603-606.
ZHOU B H, ZHOU Q, ZHENG W. Multi-factor OHT scheduling method of AMHS for semiconductor fabrications[J]. Transactions of Beijing Institute of Technology,2014,34(6):603-606.
- [2] 周炳海,李秀娟. 基于改进多目标布谷鸟搜索算法的汽车装配线物料配送调度[J]. 湖南大学学报(自然科学版),2020,47(12):1-8.
ZHOU B H, LI X J. Improved multi-objective cuckoo search algorithm for part feeding scheduling of automotive assembly lines[J]. Journal of Hunan University (Natural Science),2020,47(12):1-8.
- [3] 崔峻华,陈进,邓玉锋,等. 机械制造企业新产品试制生产任务调度方法[J]. 江南大学学报(自然科学版),2015,14(3):326-332.
CUI J H, CHEN J, DENG Y F, et al. Research for new product trial production task scheduling method in machinery manufacturing enterprises[J]. Journal of Jiangnan University (Natural Science Edition),2015,14(3):326-332.
- [4] LUO S. Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning[J]. Applied Soft Computing,2020,91:106208.
- [5] 何霆,刘飞,马玉林,等. 车间生产调度问题研究[J]. 机械工程学报,2000(5):97-102.
HE T, LIU F, MA Y L, et al. Study on shop-floor scheduling[J]. Chinese Journal of Mechanical Engineering,2000(5):97-102.
- [6] PARK J, CHUN J, KIM S H, et al. Learning to schedule job-shop problems: representation and policy learning using graph neural network and reinforcement learning[J]. International Journal of Production Research,2021,59(11):3360-3377.
- [7] 张军峰,王菲,葛腾腾. 基于分支定界法的进场航空器动态排序与调度[J]. 系统仿真学报,2016,28(8):1909-1914.
ZHANG J F, WANG F, GE T T. Dynamic arrival sequencing and scheduling based on branch and bound algorithm[J]. Journal of System Simulation,2016,28(8):1909-1914.
- [8] 赵婷,农庆琴,方奇志. 两台平行机排序博弈问题的协调机制[J]. 中国海洋大学学报(自然科学版),2013,43(7):110-114.
ZHAO T, NONG Q Q, FANG Q Z. Coordination mechanism of two parallel machines scheduling game problem[J]. Periodical of Ocean University of China (Natural Science),2013,43(7):110-114.
- [9] TASGETIREN M F, SEVKLI M, LIANG Y C, et al. A particle swarm optimization and differential evolution algorithms for job shop scheduling problem[J]. International Journal of Operations Research,2006,3(2):120-135.
- [10] ZHANG R, SONG S, WU C. A hybrid artificial bee colony algorithm for the job shop scheduling problem[J]. International Journal of Production Economics,2013,141:167-178.
- [11] 薛玲玲. 作业车间调度的块结构邻域搜索遗传算法[J]. 计算机集成制造系统,2021,27(10):2848-2857.
XUE L L. Block structure neighborhood search genetic algorithm for job-shop scheduling [J]. Computer Integrated Manufacturing Systems,2021,27(10):2848-2857.
- [12] 王艳红,赵也践,刘文鑫. 数据挖掘算法在作业车间调度问题中的应用[J]. 计算机集成制造系统,2024,30(2):520-536.
WANG Y H, ZHAO Y J, LIU W X. Application of data mining algorithm in job shop scheduling Problem[J]. Computer Integrated Manufacturing Systems,2024,30(2):520-536.
- [13] ABUALIGAH L, DIABAT A, MIRJALILI S, et al. The arithmetic optimization algorithm[J]. Computer Methods in Applied Mechanics and Engineering,2021,376:113609.
- [14] ELKASEM A H A, KHAMIES M, MAGDY G, et al. Frequency stability of AC/DC interconnected power systems with wind energy using arithmetic optimization algorithm-based fuzzy-PID controller[J]. Sustainability,2021,13(21):12095.
- [15] KHARRICH M, ABUALIGAH L, KAMEL S, et al. An improved arithmetic optimization algorithm for design of a microgrid with energy storage system; case study of El Kharga Oasis, Egypt[J]. Journal of Energy Storage,2022,51:104343.
- [16] ABUALIGAH L, ALMOTAIRI K H, ABD ELAZIZ M, et al. Enhanced Flow Direction Arithmetic Optimization Algorithm for mathematical optimization problems with applications of data clustering[J]. Engineering Analysis with Boundary Elements,2022,138:13-29.
- [17] 曹磊,叶春明,黄霞. 变邻域杂草算法在多目标柔性作业车间调度中的应用[J]. 计算机应用研究,2018,35(1):150-154.
CAO L, YE C M, HUANG X. Multi-objective flexible Job-Shop scheduling based on variable neighborhood search invasive

weed optimization[J]. Application Research of Computer, 2018, 35(1): 150-154.

- [18] LIU B, WANG L, JIN Y H. An effective PSO-based memetic algorithm for flow shop scheduling[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 2007, 37(1): 18-27.
- [19] SALIDO M A, ESCAMILLA J, BARBER F, et al. Energy efficiency, robustness, and makespan optimality in job-shop scheduling problems[J]. Artificial Intelligence for Engineering Design Analysis & Manufacturing, 2016, 30(3): 300-312.
- [20] DEVAN P A M, HUSSIN F A, IBRAHIM R B, et al. An arithmetic-trigonometric optimization algorithm with application for control of real-time pressure process plant[J]. Sensors, 2022, 22(2): 617.

Special Topics on Scheduling Theory

Improved Arithmetic Optimization Algorithm for Solving Job Shop-Scheduling Problems with Machine Aging Effects and Workpiece Release Time

CHEN Zhaohui¹, LIU Haiyang², XIA Qian³, ZHANG Xingong⁴

(1. School of Mathematics, Physics, and Data Science, Chongqing University of Science and Technology, Chongqing 401331;

2. School of Intelligent Engineering, Chongqing University of Science and Technology, Chongqing 401331;

3. Chongqing Xiejiawan School, Chongqing 400050;

4. School of Mathematical Sciences, Chongqing Normal University, Chongqing 401331, China)

Abstract: The Job Shop-Scheduling Problem (JSP), which considers the constraints of machines with aging effects and workpieces with release time, is studied. A scheduling optimization model with the objective of minimizing the maximum completion time is developed and an improved Arithmetic Optimization Algorithm (IAOA) is designed to solve the problem. The algorithm maps the IAOA continuous solution space to the discrete space of the JSP by means of ranked-order value (ROV) transformation rules, encodes the JSP and decodes it using an insertion greedy decoding algorithm. A non-linear mathematical optimization acceleration (MOA) function and six neighborhood search strategies are proposed to improve the standard Arithmetic Optimization Algorithm (AOA). The IAOA is compared with AOA, Grey Wolf Optimizer (GWO) and Arithmetic Trigonometric Optimization Algorithm (ATOA) by solving 33 benchmark problems. The experimental results show that the IAOA proposed has better optimization effect and convergence ability on JSP. The IAOA algorithm proposed overcomes the shortcomings of the AOA algorithm in terms of low solution accuracy and slow convergence speed.

Keywords: arithmetic optimization algorithm; job-shop scheduling problems; insertion greedy decoding algorithm; machine aging effect; release time

(责任编辑 黄 颖)