

# 基于混合果蝇—遗传算法求解柔性作业 车间调度问题

刘晶晶<sup>1</sup>, 刘业峰<sup>2,3</sup>, 黎虹<sup>1</sup>

1. 沈阳工学院 基础课部, 辽宁 抚顺 113122;
2. 辽宁省数控机床信息物理融合与智能制造重点实验室, 辽宁 抚顺 113122;
3. 沈阳工学院 机械工程与自动化学院, 辽宁 抚顺 113122)

**摘要:** 根据柔性作业车间调度问题的特点, 针对不同生产效率的并行设备, 以完工时间最小化为目标建立优化模型, 提出了混合果蝇优化算法和遗传算法的两阶段组合算法 (FOA-GA); 在嗅觉阶段, 通过局部路径搜索技术进行生产路径寻优; 在视觉阶段, 结合遗传算法的交叉和竞争机制, 进行个体间的信息交换, 利用寻优变异算子和常规变异算子进行两部分变异, 再引入自适应动态转移算子进行调整以加快收敛速度; 在生产实例中, 将 FOA-GA 算法与果蝇优化算法和遗传算法的结果进行比较, 证明了其可行性和有效性。

**关键词:** 果蝇优化算法; 遗传算法; 两部分变异; 自适应动态转移算子; 车间调度

## A Hybrid Algorithm of Fruit Fly Optimization Algorithm and Genetic Algorithm for Solving Flexible Job Shop Scheduling Problem

Liu Jingjing<sup>1</sup>, Liu Yefeng<sup>2,3</sup>, Li Hong<sup>1</sup>

1. Department of Basic courses, Shenyang Institute of Technology, Fushun 113122, China;
2. Liaoning Key Laboratory of Information Physics Fusion and Intelligent Manufacturing for CNC Machine, Shenyang Institute of Technology, Fushun 113122, China;
3. School of Mechanical Engineering and Automation, Shenyang Institute of Technology, Fushun 113122, China)

**Abstract:** According to the characteristics of the flexible job shop scheduling problems, an optimization model is established with the goal of minimizing the completion time, and a hybrid algorithm of fruit fly optimization algorithm and genetic algorithm (FOA-GA) is proposed. In the olfactory stage, local search technique is used to find the optimal path; In the visual stage, combining the crossover and competition mechanism of genetic algorithm, the information exchange between individuals is carried out, the mutation operator with an optimization tendency and the conventional mutation operator are used to carry out the two-part mutation, and then the adaptive dynamic transfer operator is introduced to accelerate the convergence rate. In the simulation, the results of FOA-GA algorithm are compared with those of fruit fly optimization algorithm and genetic algorithm to prove its feasibility and effectiveness.

**Keywords:** fruit fly optimization algorithm; genetic algorithm; two-part mutation; adaptive dynamic transfer operator; work-shop scheduling

## 0 引言

近年来, 智能制造技术迅速发展, 柔性作业车间的调度问题具有重要的现实意义。科学地调度其生产过程可提高

企业生产效率, 在一定程度上保障交货时间, 降低生产能耗。大量学者用不同的方法研究了自动化车间的生产调度问题。采用的方法主要有传统运筹学方法 (拉格朗日松弛方法<sup>[1]</sup>、分支定界法<sup>[2]</sup>) 等) 和智能算法 (群智能算法、人工智能算法、启发式算法等)。上述算法中, 非智能算法的求解时间复杂度较高, 并不适应大规模生产调度问题, 而智能优化算法的计算效率较高, 有时容易陷入局部极小, 需要依据问题的不同特征予以改进。

遗传算法 (GA) 是求解并行调度问题的有效算法, 目前已有大量研究。文献 [3] 根据机床的加工匹配特性, 列出 3 个不同维度的优化目标, 使用多层编码技术提出了基于遗传算法的生产调度方法; 文献 [4] 提出了求解生产调度问题的变邻域改进遗传算法; 文献 [5] 基于排列编码技术提出了一种具有并行批处理和约束的混合车间调度遗传算法; 遗传算法和其他智能优化算法相结合也是研究

收稿日期: 2020-05-26; 修回日期: 2020-06-17。

基金项目: 国家自然科学基金 (61603262); 辽宁省自然科学基金 (20180550418); 沈阳工学院 i5 智能制造研究所基金 (i5201701)。

作者简介: 刘晶晶 (1988-), 女, 辽宁铁岭人, 硕士, 讲师, 主要从事智能优化算法、生产计划和调度、人工神经网络计算等方向的研究。

黎虹 (1981-), 女, 辽宁沈阳人, 硕士, 教授, 主要从事广义系统的耗散控制方向的研究。

通讯作者: 刘业峰 (1981-), 男, 山东淄博人, 博士, 教授, 主要从事智能优化算法及应用、复杂系统建模及优化、智能制造等方向的研究。

调度问题的热点方法<sup>[6]</sup>；文献 [7] 提出多维矩阵编码技术，使用遗传算法解决了混合流水作业调度问题；文献 [8] 将传统方法与遗传算法相结合提出了基于单纯形格设计的多目标遗传算法求解车间调度问题。

其他启发式算法或群智能算法也在解决生产调度问题中发挥着重要作用。文献 [9] 提出了一种混合变邻域搜索算法 (HVNS) 来求解混合流车间调度问题，给出了八种邻域结构并提出了动态邻域更新方法提取全局信息；文献 [10] 基于生产车间调度，提出了两阶段启发式算法，第一阶段进行作业排序，第二阶段基于禁忌搜索和贪心算法的列生产算法进行抢占式调度以最小化完工时间；除此之外，粒子群优化算法<sup>[11]</sup>、人工蜂群算法<sup>[12-14]</sup>等也被广泛应用于生产调度问题中；部分调度问题将目标转向生产排产<sup>[15-20]</sup>，仍围绕生产效率最大化的主题提出了一系列启发式算法<sup>[21-22]</sup>。

2012 年，由台湾学者潘文涛提出了果蝇优化算法 (FOA)<sup>[23]</sup>，很快在优化领域受到关注。果蝇优化算法目前的研究进展<sup>[24]</sup>，包括维持种群多样性、知识驱动策略与协同机制的设计等方面的改进工作。该文献也介绍了果蝇优化算法在离散优化、多目标优化、不确定优化等方面的扩展性研究工作；求解并行调度问题时，多数学者对果蝇算法作了相应改进，包括编码方式和邻域结构生成方式的改进<sup>[25]</sup>、如何提高群体多样性以及搜索方式的改进等<sup>[26-30]</sup>。2014 年，文献 [31] 将果蝇优化算法应用到炼钢连铸的离散生产调度问题中，取得了有效的结果；2018 年，文献 [32] 提出了一种多目标果蝇优化算法 (MOFOA) 来解决测试点的选择问题；2019 年，文献 [33-34] 也成功将果蝇优化算法应用到混合流水车间调度问题中。

综上，柔性车间调度问题取得了一定的研究成果，但该调度问题属于 NP-Hard 问题，现有的优化算法还存在不足，目前将遗传算法和其它群智能优化算法结合使用或改进果蝇优化算法解决柔性作业车间调度问题的情况较多，将遗传算法和果蝇优化算法结合在一起使用较少，本文结合果蝇优化算法和遗传算法的特点，将二者结合使用，选取表达方式直接且解码方便的多维矩阵编码技术，引入寻优变异算子和自适应动态转移策略，在两阶段种群进化的基础上，以最小化完工时间为优化目标，建立了生产调度模型，得到了优化调度方案<sup>[35]</sup>。

### 1 问题描述

柔性车间生产调度问题是指多个工件在有限台设备上加工。在不同时刻，一台设备可以用于多个工件多道工序的加工，同时设备具有可选性和灵活性。因此，可以将不同设备的按序组合作为一个工件的加工路径，以达到加工用时最小化的目标。

具体描述为：加工  $m$  个工件，每个工件各有  $n$  道工序，每道工序可以在任一机器上加工，连续两道工序之间考虑运输和工序切换等待时间，各工件各道工序的加工时间均不相同。现共有  $s$  套设备，每套设备的工作效率不同，目标为使得所有工件的总加工时间最短，如图 1 所示。

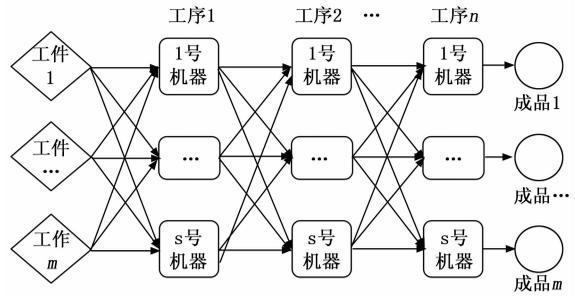


图 1 柔性车间生产线运行图

## 2 数学建模

### 2.1 符号表示

$tc_{ij}$ : 第  $i$  个工件第  $j$  道工序的加工完成时间；

$tp_{ij}$ : 第  $i$  个工件第  $j$  道工序的加工时间；

$tran_{j,j+1}$ : AGV 小车将工件物料由第  $j$  道工序到第  $j+1$  道工序之间的运输时间；

$x_{ijs}$ <sup>[1]</sup>: 0-1 变量，当且仅当工件  $i$  的第  $j$  道工序在机器号为  $s$  的机器上加工时为 1，否则为 0 ( $i=1, 2, \dots, m$ ), ( $j=1, 2, \dots, n$ ), ( $s=1, 2, 3$ )；

$y_{ikj}$ <sup>[1]</sup>: 0-1 变量，当且仅当工件  $i$  先于工件  $k$  在第  $j$  道工序被加工时为 1，否则为 0 ( $i, k=1, 2, \dots, m$ ), ( $j=1, 2, \dots, n$ )。

### 2.2 数学模型

将生产过程分解为“加工时间”和“工序间的运输时间”，通过优化总加工时间来实现生产线的高效生产，现有  $m$  个工件在  $n$  道工序上加工。

#### 2.2.1 目标函数

$$\min \sum_{i=1}^m (tc_m - tc_{i1} + tp_{i1}) \quad (1)$$

#### 2.2.2 约束条件

根据自动化生产的实际调度需求，设定如下约束条件：

1) 加工顺序约束，即同一工件在前一道工序加工完毕送至下一道工序后开始加工。

$$tc_{i,j+1} - tc_{i,j} - tp_{i,j+1} = tran_{j,j+1} \quad (i=1, 2, \dots, m; j=1, 2, \dots, n-1) \quad (2)$$

2) 机器能力约束<sup>[1]</sup>，即同一台机器不能同时加工多个工件。

$$tc_{k,j} - tc_{i,j} - tp_{k,j} + (3 - x_{ijs} - x_{kjs} - y_{ikj})U \geq 0 \quad (k \neq i, U \text{ 为较大正数}) \quad (3)$$

3) 连续加工，即一个工件加工结束后立刻加工下一个工件。

$$tc_{i+1,n} - tc_{i,n} - tp_{i+1,n} = 0, \quad i=1, \dots, m-1 \quad (4)$$

4) 各道工序加工机器唯一性约束。

$$\sum_{s=1}^3 x_{ijs} = 1, \quad i=1, 2, \dots, m, \quad j=1, 2, \dots, n \quad (5)$$

5) 工序之间的顺序关系约束。

$$y_{ijs} + y_{kjs} = 1 (k \neq i) \quad (6)$$

6) 变量约束。

$$x_{ijs} \in \{0,1\}, y_{ikj} \in \{0,1\}$$

$$(i,k = 1,2,\dots,m; j = 1,2,\dots,n; s = 1,2,3) \quad (7)$$

### 3 基于 FOA-GA 算法的求解策略

FOA 算法是基于果蝇觅食行为的生物学原理提出的一种智能优化算法。该算法分为两个阶段: 嗅觉阶段和视觉阶段。通过两个阶段的不断迭代实现觅食的优化, 得到问题的满意解。FOA 算法具有易于实现、控制参数少、便于进入问题特定的搜索环节等优点。但是, 该算法强调对个体邻域的搜索, 缺少种群多样性的变更, 因此全局搜索能力较弱, 容易陷入局部极小。

GA 算法是利用生物进化中的自然选择原理, 依据适者生存的原则构建的计算模型, 通过染色体的选择、交叉、变异方式, 进行种群的不断迭代更新, 从而找到最优解或满意解。GA 算法操作稍复杂, 但具有极大的灵活性, 对于各类特殊问题都可以按照特定方式进行求解, 可以较大程度地避免解集陷入局部极小。

本文结合柔性自动化生产线的加工调度问题, 对 FOA 算法和 GA 算法进行改进, 提出混合 FOA-GA 启发式算法, 以工件最短加工时间为目标函数, 建立了局部搜索与全局搜索相结合的搜索方式, 具体算法如下。

#### 3.1 编码方式

采用矩阵式编码结构<sup>[7]</sup>矩阵的行号代表工件编号, 矩阵的列号代表工件的加工工序, 矩阵中的元素值代表加工的机器号。对于加工工件的任意给定排序, 预先为各工件的各道工序随机分配加工机器。

$$\begin{pmatrix} mac_{11} & mac_{12} & \dots & mac_{1n} \\ mac_{21} & mac_{22} & \dots & mac_{2n} \\ \dots & \dots & \ddots & \dots \\ mac_{m1} & mac_{m2} & \dots & mac_{mn} \end{pmatrix}$$

$mac_{ij}$  表示第  $i$  个工件的第  $j$  道工序在  $mac_{ij}$  号设备上加工。另已知各工件在各道工序上的加工时间表为矩阵  $ProT$ , 其中元素可各不相同。

$$ProT = \begin{pmatrix} t_{11} & t_{12} & \dots & t_{1n} \\ t_{21} & t_{22} & \dots & t_{2n} \\ \dots & \dots & \ddots & \dots \\ t_{m1} & t_{m2} & \dots & t_{mn} \end{pmatrix}$$

#### 3.2 适应度函数以及解码方式

适应度为工件加工总时间的倒数 (即第一个工件开始加工至最后一个工件结束加工所用时间的倒数)。在矩阵编码机制下, 采用启发式算法解码并计算适应度:

1) 按工序搜索。将第  $k$  道工序中需要在机器  $m$  上加工的所有工件编号进行提取:

$$W = \{\omega_1, \omega_2, \dots, \omega_n\} \quad (8)$$

式 (8) 中,  $\omega_1, \omega_2, \dots, \omega_n$  表示在第  $k$  ( $k = 2, 3, \dots, n$ ) 道工序中第  $m$  号机器上加工的所有工件, 取出上一道工序的完工时间集合:

$$\Omega = \{tc_{\omega_1}^{k-1}, tc_{\omega_2}^{k-1}, \dots, tc_{\omega_n}^{k-1}\} \quad (9)$$

其中:  $tc_{\omega_i}^{k-1}$  表示在机器  $m$  上加工的第  $i$  个工件在第  $k-1$  道工序中的加工完成时间。

2) 排序选出在 (8) 中结束最早的工件, 将其结束时间与第  $k$  道工序中第  $m$  号机器上正在加工工件的结束时刻作比较, 二者取较大值, 将其与运输时间之和记为第  $k$  道工序中下一个工件的开始加工时间, 即:

$$tp_{next}^k = \max\{\min\{\Omega\}, tc_{current}^k\} + t_{tran} \quad (10)$$

3) 适应度为最后一个工件的加工完成时间与第一个工件开始加工时间之差的倒数。

$$f = \frac{1}{tc_{last}^n - tc_{first}^1} \quad (11)$$

### 3.3 算法流程

#### 3.3.1 种群初始化

随机产生初始种群, 并计算当前最佳适应度值, 记录初步全局最优解  $f_{cur}$ 。

#### 3.3.2 嗅觉阶段

考虑到问题结构和解的离散性, 采用多重交换 (multi-swap) 技术<sup>[31]</sup>产生每个个体的局部信息结构进行果蝇嗅觉阶段的搜索, 即在合适的范围内进行工件加工机器安排的多重交换, 在矩阵编码方式下, 其实际意义为工件加工路径的局部寻优。如图 2 所示 (假设每道工序有 5 台机器), 若交换第 1、2 行基因段, 则工件 1 每道工序对应机器的加工路径由 1, 5, 2, 3, 3 交换成 2, 4, 3, 2, 5, 这对于具有不同加工时长的各个工件寻找局部最佳加工路径具有重要作用。若当前适应度优于交换前的适应度, 则进行局部较优解以及种群个体的替换, 更新种群。

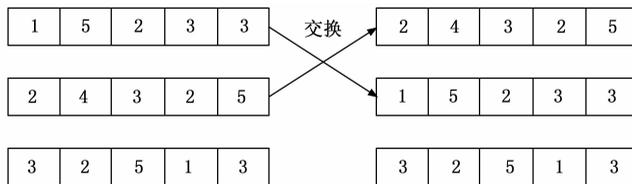


图 2 嗅觉阶段通过多重交换技术进行局部路径搜索示意图

#### 3.3.3 交叉机制

为了丰富种群的多样性, 引入遗传算法中的交叉机制。很多已有文献中采用单点交叉或两点交叉方式, 本文采用单点交叉方式, 父代两两交叉产生子代, 采用精英策略保留优势个体, 在父代和子代中采用锦标赛的方式进行优胜劣汰。矩阵编码形式下的交叉方式如图 3 所示。

#### 3.3.4 变异机制

1) 基于实际生产情况 (在每道工序 3 台机器的情况下, 1 号、2 号和 3 号机器生产效率依次降低), 选取一定比例的个体 (本文选取 80%), 设置寻优变异算法: 计算当前个体中各工序中使用 1 号设备和 3 号设备的数量  $s_1$  和  $s_3$ , 若  $s_1 > s_3$ , 变异时以  $\beta$  (本文  $\beta = 0.8$ ) 为概率, 将安排到 3 号机器生产的工件向效能较高的 1 号和 2 号机器调整, 以提高生产效率, 如图 4 所示。

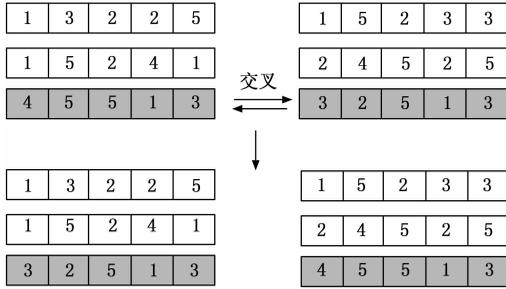


图 3 交叉操作示意图

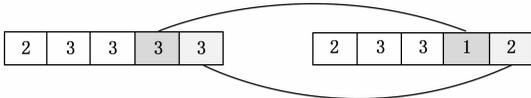


图 4 寻优变异算法操作示意图

2) 为避免陷入局部最优, 选取 20% 比例的个体, 按给定概率进行变异 (由于选取的个体较少, 本文设定变异概率为 0.3)。计算新的适应度并记录当前最佳适应度和最佳个体。

### 3.3.5 视觉阶段

整个种群向当前最优解的方向搜索。由于问题的解具有离散型, 因此常规果蝇搜索方式并不适用于此。为了不改变种群的多样性并使种群整体向好, 采用如下方法: 设置动态自适应转移比例  $\alpha(i) (i = 1, 2, \dots, iter_{max})$ 。

$$\alpha(i) = \frac{|i/3 * rand|}{N} \quad (12)$$

其中:  $N$  为种群规模。为了保持种群多样性, 在初始阶段,  $\alpha(i)$  取值较小, 随着迭代次数的增加,  $\alpha(i)$  呈现增大趋势 (但有随机扰动), 以加快收敛速度, 将适应度较差的个体以  $\alpha(i)$  为比例向最优个体调整, 以加快种群的收敛速度。

具体流程如图 5 所示。

## 4 数值实验

以某实验室智能制造车间为原型: 某生产线自动化加工单元主要由三台智能加工设备组成。做出如下假设: 车间具有三套自动化生产线, 编号为 1、2、3, 按生产效率等级分别为高、中、低。2 号设备比对应的 1 号设备加工时间多 5 个单位, 3 号设备比对应的 2 号设备加工时间多 5 个单位。需解决的问题是:  $m$  (一般为 8 的整数倍) 个工件在生产线上按顺序进行  $n$  个工序的加工, 工件可以由 AGV 小车运输调度, 在对应工序的任一设备上加工, 不考虑第一道工序开始前的物料运输时间, 如何使得总加工时间最短。

以 16 个工件、3 道工序为例, 设定各工件在各道工序加工所需时间如表 1 所示, AGV 小车在各道工序之间的运输时间如表 2~3 所示。

例如, 表 2 中 4.4387 表示工件由第一道工序的 1 号机器运输到第二道工序的 1 号机器所需时间。

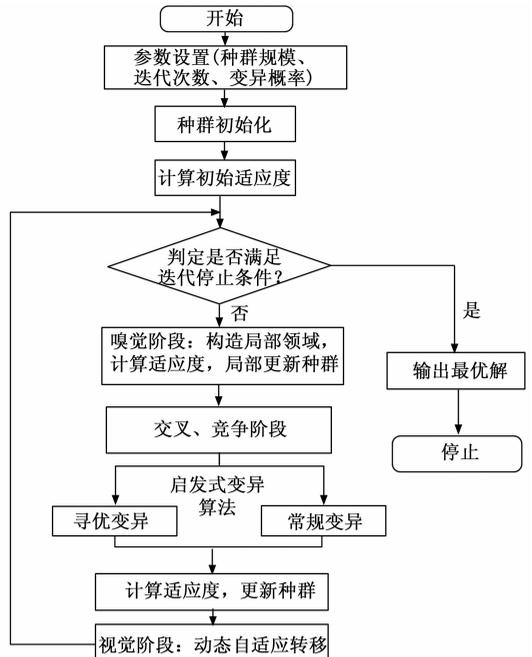


图 5 FOA-GA 算法流程图

表 1 工件在各道工序的加工时间表 (以 1 号机器为标准)

工件编号 \ 工序	1	2	3
1	49.555 1	37.720 2	48.996 8
2	49.993 8	47.031 2	42.143 6
3	38.885 6	43.333 6	45.814 2
4	39.168 1	47.008 4	38.367 7
5	47.389 0	44.596 4	36.282 5
6	49.584 1	43.708 5	43.922 9
7	36.572 4	36.752 7	39.552 7
8	45.312 0	40.637 6	41.539 8
9	38.535 6	36.263 7	39.623
10	45.197 7	48.181 9	45.800 8
11	49.886 2	46.854 8	46.314 4
12	46.692 9	44.069 1	38.759 4
13	39.305 3	48.412 8	40.444 9
14	42.909 5	47.719 9	46.401 0
15	38.511 9	46.403 1	43.161 0
16	43.937 5	39.082 7	36.462 1

表 2 第 1-2 工序之间 AGV 小车运输时间表

机器编号	1	2	3
1	4.438 7	4.791 8	5.501 0
2	9.725 7	8.769 2	3.273 5
3	3.597 3	3.267 1	8.183 6

表 3 第 2-3 工序之间 AGV 小车运输时间表

机器编号	1	2	3
1	9.434 8	9.331 4	3.650 3
2	8.277 3	6.400 1	3.777 1
3	3.852 5	3.750 0	4.172 1

实验参数设置如表 4 所示。

表 4 参数设置表

参数名称	值
种群规模	200
最大迭代次数	200
加工工件个数	16
工序个数	3-5
每道工序可用机器数量	3
多重交换技术随机数范围	[1,9]
寻优变异比例及概率	80%, 0.3
常规变异比例及概率	20%, 0.3

为了排除实验结果的随机性因素, 现给出 10 次实验结果的平均值和最优值, 在相同参数设置下分别对比 FOA-GA、GA 和 FOA 算法的实验结果。使用 MATLAB R2019a 编写程序, 在 8 G 内存、1.6 GHz 的 CPU 上运行, 实验结果如表 5 所示。

表 5 三种算法最优加工时间平均值对比表 (16 个工件)

实验次序	FOA-GA	GA	FOA
1	399. 386 8	410. 817 5	428. 737 1
2	416. 536 7	426. 631 3	427. 763 2
3	413. 794 1	412. 970 6	412. 007 9
4	413. 445 0	421. 983 8	422. 145 6
5	403. 524 5	420. 204 2	419. 076 5
6	398. 995 3	421. 660 9	418. 230 8
7	403. 745 8	413. 508 9	419. 887 1
8	413. 532 8	416. 088 7	416. 746 3
9	413. 780 6	420. 129 9	424. 343 7
10	405. 106 4	420. 876 4	412. 913 9
平均值	408. 1848	418. 4872	420. 285 2
最优值	398. 9953	410. 8175	412. 007 9

由表 5 可知, 在加工 16 个工件时, FOA-GA 算法、GA 算法和 FOA 算法的最优解 (即最短加工时间) 分别为 398.995 3、410.817 5 和 412.007 9 比较其平均值, FOA-GA 算法得到的工件加工时间比 GA 算法平均节约 10.302 4 个单位, 比 FOA 算法平均节约 12.100 4 个单位, 证明了算法的有效性。

现给出在 FOA-GA 算法下取得最优解时的调度方案。

图 6 中, “1-1” 代表第 1 个工件的第 1 道工序, 机器编号 1、4、7 分别对应第 1 道工序的 1-3 号机器, 机器编号 2、5、8 分别对应第 2 道工序的 1-3 号机器, 机器编号 3、6、9 分别对应第 3 道工序的 1-3 号机器。当 3 套设备之间的生产效率差距较大时, 寻优算子的作用更加明显, 机器分配的合理性更加显著。FOA-GA 算法的收敛曲线如图 7~8 所示。

如图 7 所示, 使用 FOA-GAG 算法, 经过 27 次迭代后, 曲线收敛, 工件加工时间的最优解为 398.995 3 个时间单位。为了体现视觉阶段动态自适应转移算子的效果, 图 8

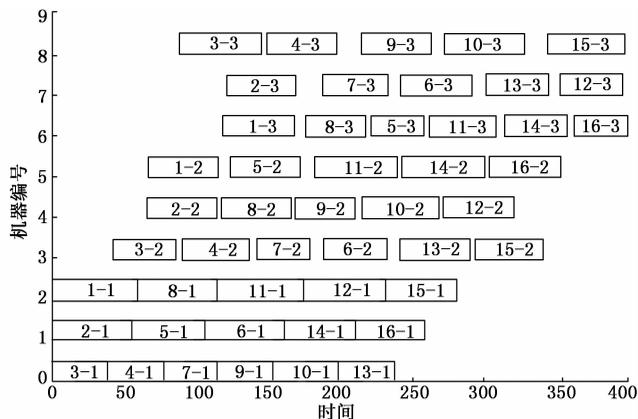


图 6 加工 16 个工件的生产任务甘特图 (FOA-GA 算法)

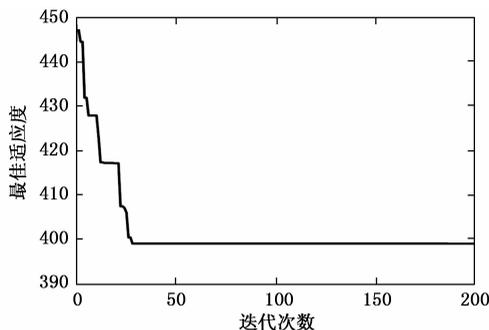


图 7 引入动态自适应转移算子后的最优解收敛曲线图

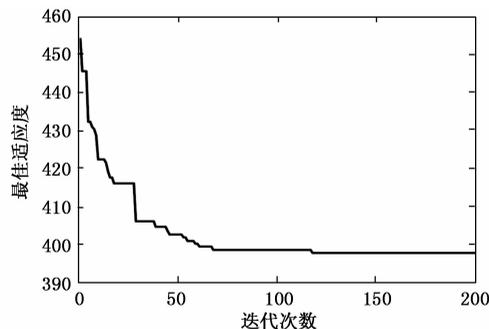


图 8 未引入动态自适应转移算子时的最优解收敛曲线图

给出了在近乎相同最优解下引入动态自适应转移算子前后的实验结果对比, 可见引入动态自适应算子后, 最优解曲线的收敛时间由第 118 次迭代步数加速至第 27 次迭代步数, 效果显著。

采用 GA 算法和 FOA 算法取得最优解时的收敛曲线如图 9~10 所示。

由图 9 可知, GA 算法在经过 74 次迭代后取得最优解为 410.817 5, 比 FOA-GA 算法多耗时 11.822 2 个时间单位。从图 10 可以看出, FOA 算法在迭代 140 次后取得最优解 412.007 9, 比 FOA-GA 算法多耗时 13.012 6 个时间单位, 且 FOA 算法收敛较慢。以上对比证明了 FOA-GA 算法的优越性和有效性。

实际生产中, 工件数量较大, 现分别考虑加工 40、80、

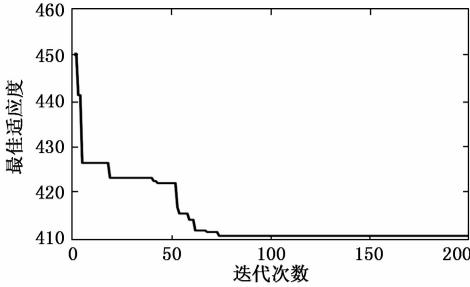


图 9 最优解收敛曲线图 (GA)

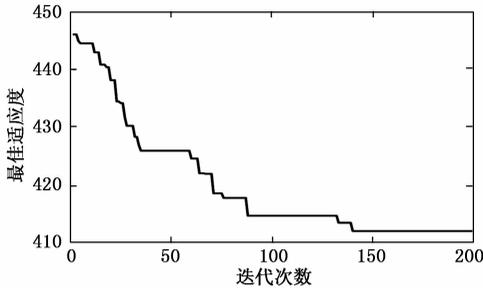


图 10 最优解收敛曲线图 (FOA)

120、200 个工件时的情况，并将加工 16 个工件时的实验结果也记录于表 6 中。在每次实验中，工件在各道工序的加工时间随机生成在 [36, 50] 之间，工序间的转移运输时间随机生成在 [3, 10] 之间，比较 10 次实验的平均值。

表 6 3 种算法的加工时间对比表(3 道工序、不同工件数)

工件个数	FOA-GA	GA	FOA
16	408.2	418.5	420.3
40	839.1	864.9	877.2
80	1 584.3	1 621.8	1 634.7
120	2 318.8	2 325.8	2 378.9
200	3 809.3	3 838.0	3 927.0

由表 6 可以看出，分别加工 16、40、80、120、200 个工件时，FOA-GA 算法均表现出优势所在。例如，当加工 80 个工件时，FOA-GA 算法所得的最优解为 1 584.3，工件的加工用时比 GA 算法平均节约 37.5 个单位，比 FOA 算法平均节约 50.4 个单位；当加工 200 个工件时，FOA-GA 算法的最优解平均值为 3 809.3，加工用时比 GA 算法平均节约 28.7 个单位，比 FOA 算法平均节约 117.7 个单位。

实际加工中，往往不只 3 道工序，现给出在加工 16 个工件、工序个数分别为 3、4 和 5 时 3 种算法在 10 次实验中平均值对比结果，如表 7 所示。

表 7 3 种算法的加工时间对比表(16 个工件、不同工序数)

工序数	FOA-GA	GA	FOA
3	399.0	410.8	412.0
4	486.1	495.0	496.7
5	563.4	563.9	595.2

算法的结果对比，可以看出 FOA 算法均占有优势，再一次验证了算法的有效性。

### 5 结束语

本文在结合果蝇优化算法 (FOA) 和遗传算法 (GA) 的基础上，通过矩阵编码机制，引入局部加工路径搜索技术和以提高生产效率为原则的寻优变异算子，解决了工件自动化生产调度问题，再通过自适应动态转移算子，加快了算法的收敛性。通过数值实验，横向对比 GA 算法和 FOA 算法，证明了 FOA-GA 算法的优越性和有效性。将上述算法依据企业的智能车间进行改进，便可进行实际应用。

#### 参考文献:

- [1] 毛 坤, 潘全科, 庞新富. 求解炼钢—连铸生产调度问题的拉格朗日算法 [J]. 系统工程学报, 2014, 29 (2): 233-245.
- [2] Bellabdaoui A, Teghem J. A mixed—integer linear programming model for the continuous casting planning [J]. International Journal of Production Economics, 2006, 104 (2): 260-270.
- [3] 孔 琳, 王黎明, 李方义, 等. 基于机床加工匹配特性的混合流水车间绿色生产调度 [J]. 计算机集成制造系统, 2019, 25 (5): 1075-1085.
- [4] 崔 琪, 吴秀丽, 余建军. 变邻域改进遗传算法求解混合流水车间调度问题 [J]. 计算机集成制造系统, 2017, 23 (9): 1917-1927.
- [5] Costa A, Cappadonna F A, Fichera S. A novel genetic algorithm for the hybrid flow shop scheduling with parallel batching and eligibility constraints [J]. International Journal of Advanced Manufacturing Technology, 2014, 75 (5/6/7/8): 833-847.
- [6] Mousavi M, Yap H J, Musa S N, et al. Multi—objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization [J]. PLoS One, 2017, 12 (3): e0169817. DOI: 10.1371/ journal. Pone.0169817.
- [7] 王万良, 姚明海, 吴云高, 等. 基于遗传算法的混合 Flow—shop 调度方法 [J]. 系统仿真学报, 2002, 14 (7): 863-865.
- [8] Jiang Y L, Li X Y, Gao L, et al. Energy—efficient job shop scheduling problem with variable spindle speed using a novel multi—objective algorithm [J]. Advances in Mechanical Engineering, 2017, 9 (4): 1-21.
- [9] Li J Q, Pan Q K, Wang F T. A hybrid variable neighborhood search for solving the hybrid flow shop scheduling problem [J]. Applied Soft Computing, 2014, 24: 63-77.
- [10] Figielska E. A heuristic for scheduling in a two—stage hybrid flowshop with renewable resources shared among the stages [J]. European Journal of Operational Research, 2014, 236 (2): 433-444.
- [11] Zarrouk R, Bennour I E, Jemai A. A two—level particle swarm optimization algorithm for the flexible job shop scheduling problem [J]. Swarm Intelligence, 2019, 13 (3): 145-168.