

# 分布式估计算法在考虑差异工件的并行批处理机调度中的应用<sup>①</sup>



张 建

(中国科学技术大学 管理学院, 合肥 230026)

通讯作者: 张 建, E-mail: [flyuii@mail.ustc.edu.cn](mailto:flyuii@mail.ustc.edu.cn)

**摘 要:** 论文考虑包含差异工件的并行批处理机调度问题, 优化目标是 minimized 制造跨度. 在不违背机器容量的限制下, 所有工件需要被分成不同的批次, 然后被安排在机器上进行加工. 首先根据问题提出一个混合整数规划模型, 并提出一个下界; 采用 FF-LPT 规则实现对工件的分批和排序; 然后提出基于 4 种更新机制的分布式估计算法 (EDA) 来对问题求解. 最后通过实验对各类规模不同的算例进行仿真, 并将结果和模拟退火算法 (SA)、遗传算法 (GA) 作对比, 验证了算法的有效性.

**关键词:** 批调度; 并行批处理机; 分布式估计算法; 差异工件

引用格式: 张建. 分布式估计算法在考虑差异工件的并行批处理机调度中的应用. 计算机系统应用, 2019, 28(6): 213-220. <http://www.c-s-a.org.cn/1003-3254/6916.html>

## Application of Estimation of Distribution Algorithm to Parallel Batch Processing Machines with Non-Identical Job Sizes

ZHANG Jian

(School of Management, University of Science and Technology of China, Hefei 230026, China)

**Abstract:** This paper aims at minimizing makespan of parallel batch processing machines with non-identical job sizes. All the jobs are grouped into batches within the restraint of machines' capacity and then scheduled on the machines. First, a mixed integer programming model is summarized for this problem and a lower bound is proposed. Then an FF-LPT rule is addressed to form batches and assign batches on machines. And an estimation of distribution algorithm (EDA) with four different update mechanism is proposed. The performance of proposed algorithm is evaluated by comparing with a simulated annealing algorithm (SA) and a genetic algorithm (GA). The experimental results indicate that the effectiveness of proposed algorithm.

**Key words:** batch scheduling; parallel batch processing machines; estimation of distribution algorithm; non-identical job sizes

### 引言

在工业生产中, 集成电路芯片的制造包括生产、探测、装配和测试环节<sup>[1-3]</sup>. 在测试过程中, 需要将半导体芯片置于高温烤箱中进行耐温测试以检测出潜在

的风险. 为了提高测试效率, 芯片被分成不同的批次集中放在烤箱中处理. 这里, 半导体芯片就类似于工件, 烤箱类似于机器. 这是一个典型的批处理机调度的例子.

批调度问题不同于传统的机器加工问题, 一个批

<sup>①</sup> 基金项目: 国家自然科学基金 (71671168)

Foundation item: National Natural Science Foundation of China (71671168)

收稿时间: 2018-12-05; 修改时间: 2018-12-26; 采用时间: 2019-01-03; csa 在线出版时间: 2019-05-25

处理机能同时加工处理多个工件,从而有效提高资源的利用率.研究批调度问题,通过优化调度有效提高生产效率,降低生产成本,对于提高生产管理水平、获取更高的经济效益有着重要的现实意义<sup>[4]</sup>.

人们对批调度问题的研究由来已久. Gimple 和 Ikura<sup>[5]</sup>最先研究了工件尺寸相同、到达时间和交货期相一致的单机批调度问题. Lee 和 Uzsoy<sup>[6]</sup>提出了动态规划的方法来解决此类调度问题,优化目标是最大化最大延迟时间和最小化延迟工件数,局限性在于这种基于动态规划的精确求解方式需要较长的计算时间. Uzsoy<sup>[7]</sup>首次将工件尺寸相同的性质扩展到差异工件. 在此基础上, Zhou<sup>[8]</sup>融入了工件动态到达的限制,并结合距离矩阵提出了构造性启发式算法,这些一次性启发式算法在处理大规模工件问题时虽然在运行时间上占优,但解的质量并不太好.

在实际生活中,工厂往往采用多台机器并行处理的方式来提高生产效率. Chang<sup>[9]</sup>、Damodaran 和 Chang<sup>[10]</sup>等考察了并行批处理机调度问题,优化目标是最大化制造跨度. 其中 Chang 采用模拟退火算法,通过交换初始解中不同批中的工件生成邻域解,这导致初始解的质量对最终解的质量有很大影响,从而影响算法的性能. Chen<sup>[11]</sup>在并行机基础上纳入了动态到达的约束,并结合蚁群算法和遗传算法以及 ERT-LPT 规则来优化完工时间,并且取得了不错的效果.

元启发式算法也被越来越多的学者应用到调度问题中. 例如基于构建性的蚁群算法<sup>[12]</sup>,基于邻域搜索的模拟退火算法<sup>[3,9]</sup>和进化算法,其中进化算法包含但不限于遗传算法<sup>[13]</sup>、粒子群算法<sup>[14]</sup>、差分进化算法<sup>[15]</sup>等. 分布式估计算法作为一种较新的基于种群的进化算法,在求解组合优化问题方面具有良好的表现,比如置换流水车间问题<sup>[16]</sup>,护士排班问题<sup>[17]</sup>,但是在并行机调度方面的应用研究却很有限. 因此,本论文采用分布式估计算法求解考虑差异工件的并行批处理机调度问题.

论文组织如下:第1章节为问题描述;第2章节详细介绍分布式估计算法在考虑差异工件的并行批处理机调度中的应用;第3章节为仿真实验;第4章节为结论与展望.

## 1 问题描述

### 1.1 数学模型

论文考虑包含差异工件的并行批处理机调度问题,

相关假设如下:

- (1) 将  $n$  个工件分配到  $m$  个批处理机加工;
- (2) 工件具有不同的尺寸和加工时间;
- (3) 各台机器容量一致,工件  $j$  只需在其中一台机器上加工,且它在所有机器上加工时间相同;
- (4) 工件集中分批,按照批次顺序安排在机器上加工;
- (5) 批的尺寸等于批中所有工件尺寸之和,且不能超过机器容量限制,批的加工时间等于批中工件最大加工时间;
- (6) 一旦某个批次开始加工,加工过程不能中断且不能有新的工件加入到当前批中;
- (7) 优化目标是 makespan,也就是使最后离开加工系统的批次的完工时间最小;

该问题采用调度理论中的三元组法<sup>[18]</sup>可表示为  $P_m|batch, B, s_j|C_{max}$ , 以下给出相关参数定义及混合整数规划模型:

#### ① 集合

工件集合:  $\{j | j = 1, 2, 3, \dots, n\}$

批集合:  $\{b | b = 1, 2, 3, \dots; b \leq n\}$

机器集合:  $\{k | k = 1, 2, 3, \dots, m\}$

#### ② 参数

$s_j$ : 工件  $j$  尺寸;

$p_j$ : 工件  $j$  加工时间;

$B$ : 机器容量;

#### ③ 决策变量

$PT_{bk}$ : 在机器  $k$  上加工的批  $b$  的加工时间;

$CT_{bk}$ : 批  $b$  在机器  $k$  上的完工时间;

$C_{max}$ : 最终完工时间;

$x_{jbk}$ : 工件  $j$  分在批  $b$  中安排在机器  $k$  上加工时为 1, 否则为 0;

数学模型如下:

$$\text{Minimise } C_{max} \quad (1)$$

$$\text{s.t. } \sum_{b=1}^n \sum_{k=1}^m x_{jbk} = 1, \quad j = 1, \dots, n \quad (2)$$

$$\sum_{j=1}^n s_j x_{jbk} \leq B, \quad b \leq n; k = 1, \dots, m \quad (3)$$

$$PT_{bk} \geq p_j x_{jbk}, \quad j = 1, \dots, n; b \leq n; k = 1, \dots, m \quad (4)$$

$$CT_{bk} = CT_{b-1,k} + PT_{bk}, \quad b \leq n; k = 1, \dots, m \quad (5)$$

$$C_{\max} \geq CT_{bk}, \quad b \leq n; k = 1, \dots, m \quad (6)$$

$$x_{jbk} = \{0, 1\}, \quad j = 1, \dots, n; b \leq n; k = 1, \dots, m \quad (7)$$

式(1)为优化目标;式(2)保证一个工件只能分配在一个批中只能在一个机器上加工;式(3)保证批的容量不超过机器的容量限制;式(4)说明批的加工时间为批中最长加工时间;式(5)表明前后批完工时间之间的关系;式(6)为  $C_{\max}$  的计算方法;式(7)是对变量  $x_{jbk}$  的取值约束。

## 1.2 问题下界

为了能更有效评估后续算法中解的质量,我们考虑这样一个下界:假设存在某种单位工件  $e_j$  具有单位尺寸单位工时,一个尺寸为  $s_j$ 、加工时间为  $p_j$  的普通工件可表示为  $s_j * p_j$  个  $e_j$ , 此时问题转化成若干个性质相同的单位工件的并行机调度问题。理想状态下所有单位工件能完美分配到批中,即批中所有工件尺寸之和恰好等于批的容量,然后均匀分配到各台机器上加工,可得:

$$LB = \sum_{i=1}^n s_i p_i / (m * B)$$

显然这样计算得出的结果是一个有效的下界,其中批的利用率达到最高。

## 2 分布式估计算法在考虑差异工件的并行批处理机调度中的应用

### 2.1 分布式估计算法

分布式估计算法是一种基于概率模型的进化算法,最早于1996年由 Mühlenbein, Bendisch<sup>[19]</sup>等人提出,是当前国际进化计算领域的研究热点。根据变量之间的相关性,算法可分为三类:变量无关、双变量相关和多变量相关;按照变量的类型可分为离散变量的分布式估计算法和连续域上的分布式估计算法。

分布式估计算法不同于经典的遗传算法,它摒弃了遗传算法中采用选择、交叉、变异的方式生成子代种群的策略,转而采用概率模型来更新下一代种群,通过收集统计全局信息,学习优势个体特征来更新概率模型进行采样,然后生成新的个体。该算法基本流程如图1所示。

### 2.2 编码与解码

论文采用基于工件序列的编码方式,每个工件有相应的序列编号,根据不同的排列可得到不同的调度

方案。

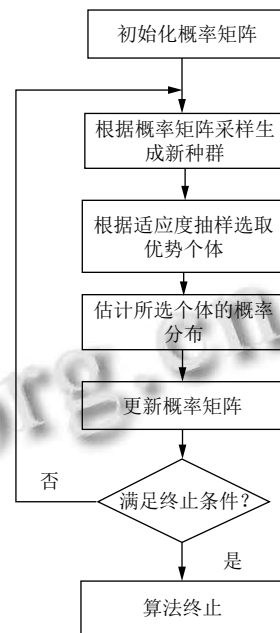


图1 EDA基本流程图

给定一个工件序列,首先我们采用FF(First-Fit)规则进行分批,然后根据LPT(Longer Processing Time)方法选择当前处于空闲状态的机器进行调度。FF规则如下:

步骤1:选择工件序列中第一个工件加入到当前批中,若违反机器容量约束,考虑下一个工件;

步骤2:每当工件完成分批,则从序列中去除该工件;若当前批容量已满或不能容纳其它任何工件,则新建一个批;

步骤3:重复步骤1~2,直到所有工件完成分批;

分批完成后,按批次的加工时间由长到短重新排列,并根据此排列顺序选择非加工状态的机器进行加工,若存在多台非加工状态的机器,则任选其中之一进行加工。

考虑10个工件在两台机器上加工,假设各台机器容量均为15,工件序列为[4,5,1,3,6,2,9,10,7,8],工件属性及分批调度结果如图2所示。

### 2.3 概率模型与更新机制

算法采用概率矩阵  $P$  来表征分布估计模型,  $P_{ij}(t)$  表示第  $t$  代中工件  $i$  出现在位置  $j$  周围的概率,这里用“周围”是因为我们考虑了几种不同的更新策略来计算工件  $i$  在位置  $j$  左右的概率,而不是仅仅在固定位置  $j$  上。

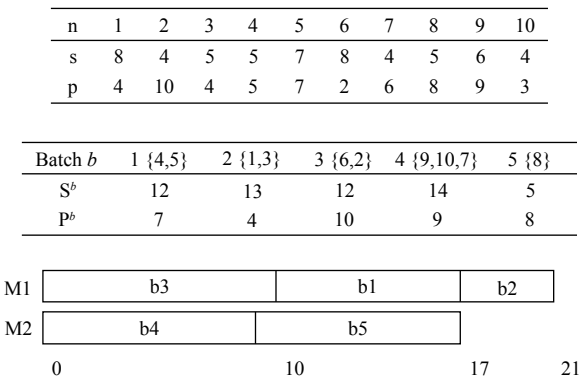


图2 10工件调度示意图

$$P_{ij}(t) = \begin{Bmatrix} p_{11}(t) & \cdots & p_{1n}(t) \\ \vdots & \ddots & \vdots \\ p_{n1}(t) & \cdots & p_{nn}(t) \end{Bmatrix}$$

算法初始阶段所有概率值均设为  $1/n$ , 以保证工件能被均匀抽样. 对于每一代种群, 设种群规模为  $Q$ , 我们根据个体的  $C_{max}$  值计算适应度值 (*fitness*), 然后挑选最优  $SP\_Size$  个个体估计其概率分布, 通过对其学习来更新概率矩阵. 这里的 *fitness* 及  $SP\_Size$  计算如下:

$$fitness(i) = 1.1 * \max\{C_{max}(1), \dots, C_{max}(Q)\} - C_{max}(i);$$

$$SP\_Size = \alpha * Q$$

其中,  $\alpha$  为优势个体选择比例,  $0 < \alpha < 1$ .

论文中采用 FF-LPT 规则解码, 因此工件的位置对于批的形成有很大影响, 相邻位置的工件就越有可能分配到同一个批中. 我们考虑 4 种更新机制来完善概率矩阵, 第一种更新机制学习优势个体中当前位置工件的概率分布; 第二种更新机制学习当前位置及其之前位置工件的概率分布; 第三种更新机制与第二种相反, 通过学习优势个体中当前位置及其之后位置工件的概率分布; 第四种更新机制考虑当前位置及其邻域内工件的概率分布. 具体更新方式如下:

$$(1) p_{ij}(t+1) = (1-\beta)p_{ij}(t) + \frac{\beta}{SP\_Size} \sum_{k=1}^{SP\_Size} I_{ij}^k$$

$$I_{ij}^k = \begin{cases} 1, & \text{if 工件 } i \text{ 放在位置 } j \text{ 上} \\ 0, & \text{otherwise} \end{cases}$$

$$(2) p_{ij}(t+1) = (1-\beta)p_{ij}(t) + \frac{\beta}{j * SP\_Size} \sum_{k=1}^{SP\_Size} I_{ij}^k$$

$$I_{ij}^k = \begin{cases} 1, & \text{if 工件 } i \text{ 放在位置 } j \text{ 或之前位置} \\ 0, & \text{otherwise} \end{cases}$$

$$(3) p_{ij}(t+1) = (1-\beta)p_{ij}(t) + \frac{\beta}{(n-j+1) * SP\_Size} \sum_{k=1}^{SP\_Size} I_{ij}^k$$

$$I_{ij}^k = \begin{cases} 1, & \text{if 工件 } i \text{ 放在位置 } j \text{ 或之后位置} \\ 0, & \text{otherwise} \end{cases}$$

$$(4) p_{ij}(t+1) = (1-\beta)p_{ij}(t) + \frac{\beta}{SP\_Size} \sum_{k=1}^{SP\_Size} \frac{I_{ij}^k}{|v_j|}$$

$$I_{ij}^k = \begin{cases} 1, & \text{if 工件 } i \text{ 放在位置 } j \text{ 邻域范围内,} \\ & \text{邻域大小为 } |v_j| \\ 0, & \text{otherwise} \end{cases}$$

其中,  $\beta$  为学习率,  $0 < \beta < 1$ ;  $v_j$  为单边邻域大小, 例如当  $v_j=2$  时, 位置 5 的邻域为位置 3, 4, 5, 6, 7,  $|v_5|$  为 5; 位置 1 的邻域为位置 1, 2, 3,  $|v_1|$  为 3.

为了阐明概率矩阵更新过程, 我们用一个包含 5 个工件的算例示范. 假设挑选 4 个优势个体, 如图 3 所示. 设  $v_j$  为 1, 根据 4 种更新机制统计各个位置的工件信息, 构建优势个体的概率分布矩阵, 结果如图 4 所示. 通过学习优势个体的概率分布特征来更新概率矩阵  $P$ , 然后基于比例选择采用轮盘赌的方式依次决定各个位置上工件编号, 即可获得新的个体.

个体1	2	1	5	4	3
个体2	3	2	1	5	4
个体3	4	5	3	2	1
个体4	3	4	2	5	1

图3 挑选的 4 个优势个体

### 2.4 面向差异工件的并行批处理机调度问题的分布式估计算法

综上, 面向差异工件的并行批处理机调度问题的分布式估计算法如算法 1 所示.

算法 1. 面向差异工件的并行批处理机调度问题的分布式估计算法

```

初始化参数  $Q, \alpha, \beta, gen$ (迭代次数), 随机生成初始种群;
 $t = 0$ ;
While  $t < gen$ 
    采用 FF-LPT 规则对种群中个体进行解码, 计算适应度值;
    根据适应度值选取最优的  $SP\_Size$  个个体, 并估计其概率分布;
    选取 2.3 中的更新策略对概率矩阵进行更新;
    根据新的概率矩阵采样生成新的种群;
    令  $t = t + 1$ ;
end
输出最优调度方案及 makespan.
    
```

### 3 实验设计与结果分析

#### 3.1 实验设计

本实验根据文献[9]提出的方法生成随机算例, 算例规模按 3 个标准划分: 工件数量  $n$ , 工件尺寸  $s$ , 加工时间  $p$ ; 其中  $s$ 、 $p$  均服从离散均匀分布; 机器数量

$m$  为 2 或 4, 容量均为 20. 具体参数见表 1.

每类算例提供一个编号. 例如包含 50 个工件, 工件尺寸服从[1, 10]分布, 加工时间满足[1, 20], 在两台机器上加工的一类算例编号为 J2S3P2M1. 每类算例随机生成 10 个子算例, 每个算例计算 10 次.

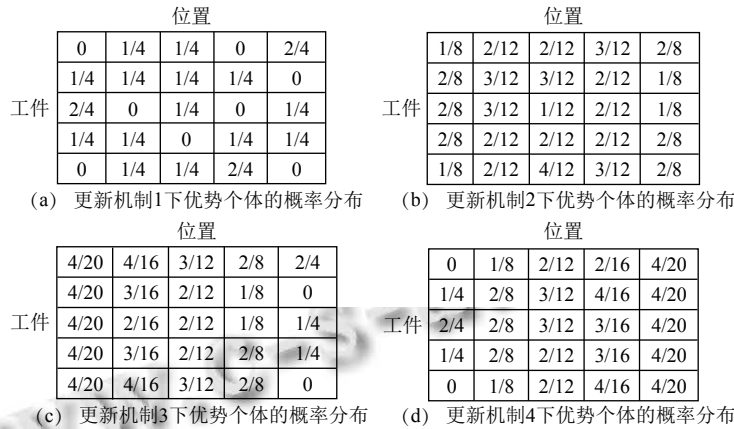


图 4 不同更新机制下优势个体概率矩阵计算结果

表 1 算例规模

维度	等级
$n$	20, 50, 100
$s$	Uniform [2,4], [4,8], [1,10]
$p$	Uniform [1,10], [1,20]
$m$	2, 4

影响 EDA 性能的参数主要有 4 个:  $Q, \alpha, \beta, gen$ . 其中, 优势个体的选择比例  $\alpha$  对概率矩阵的更新有很大影响, 若比例选择过小, 子代种群与当代个体相似度高, 则容易陷入局部最优; 若比例选择过大, 个体的优势特征弱化, 导致概率矩阵的学习效果差. 为此, 论文考虑 3 种不同的  $\alpha$  值进行预实验. 为了选取合适的参数值展开后续实验, 我们采用田口方法对不同水平的参数进行评估, 将参数作为因子, 并对每个因子设置 3 水平的参考值, 具体的参数设置见表 2.

表 2 参数设置

参数	水平		
	1	2	3
$Q$	40	50	60
$\alpha$	0.1	0.2	0.3
$\beta$	0.1	0.2	0.3
$gen$	200	300	500

算法提供了 4 种更新策略, 为方便表述, 依次记为 EDA1, EDA2, EDA3, EDA4, 其中 EDA3 表示采取第三

种更新策略. 由于 EDA4 中需要考虑单边邻域  $v$ , 因此为 EDA4 单独设立 5 因素 3 水平的实验, 参数顺序为  $Q, \alpha, \beta, v, gen$ .  $v$  具有 3 水平, 分别是[2,3,5], 其余因素同表 2.

根据实验参数和因子水平, 选取正交阵列  $L9(3^4)$  和  $L27(3^5)$ , 其中  $L9(3^4)$  表示针对 3 因素 4 水平的问题设计 9 种实验. 我们针对 50 规模的工件, 生成  $12(3 \times 2 \times 2)$  个算例, 每个算例采取 9 次试验, 算 5 次, 一共计算  $12 \times 9 \times 5$  次; 对于 EDA4, 需要计算  $12 \times 27 \times 5$  次. 最后的平均响应变量值 (ARV) 计算方式如下:

$$ARV = \frac{\sum_{i=1}^n C_{\max}(i)/LB(i)}{n}$$

其中,  $n$  为算例个数. 显然, ARV 越小越好.

图 5 中各子图横坐标为因子水平值, 纵坐标为 ARV, 分析可知, EDA1 最优参数组合为:  $Q=60, \alpha=0.2, \beta=0.1, gen=500$ ; EDA2 最优组合为:  $Q=60, \alpha=0.1, \beta=0.1, gen=500$ ; EDA3 最优组合为:  $Q=50, \alpha=0.1, \beta=0.3, gen=500$ ; EDA4 最优组合为:  $Q=60, \alpha=0.1, \beta=0.3, v=2, gen=500$ .

#### 3.2 实验结果及分析

为了评估算法的性能, 我们将和文献[9]提出的模拟退火算法 (SA) 以及经典的遗传算法 (GA) 作对比. SA 中各项参数与原文献一致, GA 中种群规模和迭代

次数同 EDA 一样, 变异概率设为 0.1. 所有的算法均由 Matlab 实现, 实验环境为 4 核 3.4 GHz、8 GB RAM 的 Windows10 系统.

表 3、表 4 分别给出了 2 台和 4 台机器环境下所有算例的结果. 实验中分别统计了各个算法运行得到

的最优值、平均值、最差值及运行时间等, 这里只给出了比值 *ratio*, 计算公式为:

$$ratio = ave(C_{max})/LB$$

*ratio* 越小, 说明结果越接近下界, 所得到的解越接近最优解.

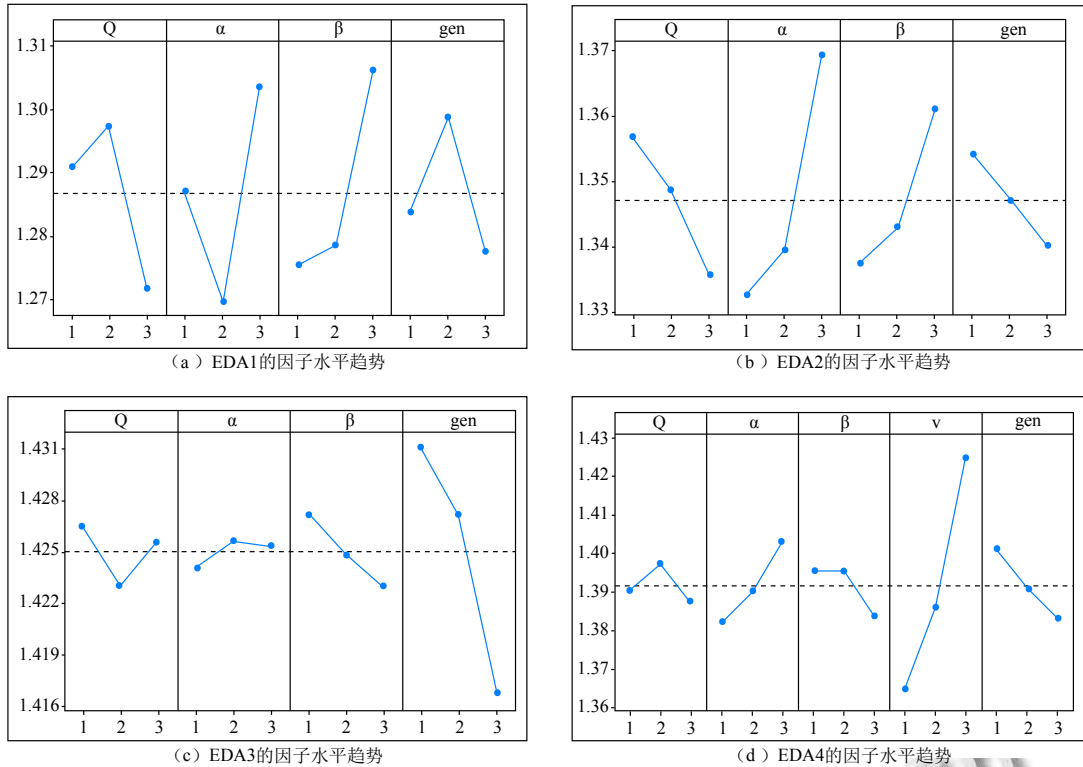


图 5 不同更新机制下 EDA 的因子水平趋势

从表 3、表 4 中可以看出, 对于绝大部分算例, 我们提出的 4 种 EDA 得到的解均是要优于 SA 算法的. 对于 2 台机器, EDA1 的 *ratio* 均值为 1.24, GA 为 1.27, SA 为 1.44. 随着工件规模的增加, 算法的解的质量也越来越好, 尤其是在较多并行机的情况下.

值得注意的是, 4 种 EDA 算法中, EDA1 的性能是最好的, 其次是 EDA2, EDA3 的性能最差. EDA2 与 EDA3 的更新机制是截然相反的, EDA3 考虑当前位置及其之后位置工件的概率, 但是由于采样时我们是按照顺位比例选择的方式依次决定各个位置的工件, 这导致 EDA3 效率较低. 而 EDA1 只考虑当前位置的工件的概率分布, 使得概率分布极为“精纯”, 所以性能较好.

另外, GA 的性能比想象中好很多, 对于小规模算例, GA 的解甚至比 EDA1 还要占优; 但对于大规模工件, EDA1 得到的解的质量要优于 GA.

表 3 M1 类算例结果

算例	算法					
	EDA1	EDA2	EDA3	EDA4	GA	SA
J1S1P1	1.40	1.40	1.49	1.40	1.37	1.63
J1S1P2	1.44	1.42	1.50	1.43	1.40	1.61
J1S2P1	1.21	1.20	1.25	1.23	1.20	1.32
J1S2P2	1.24	1.23	1.31	1.27	1.21	1.41
J1S3P1	1.19	1.19	1.24	1.20	1.19	1.35
J1S3P2	1.21	1.20	1.26	1.22	1.19	1.35
J2S1P1	1.28	1.38	1.52	1.40	1.28	1.54
J2S1P2	1.26	1.35	1.49	1.39	1.29	1.50
J2S2P1	1.16	1.20	1.34	1.26	1.22	1.38
J2S2P2	1.18	1.23	1.36	1.28	1.24	1.39
J2S3P1	1.16	1.18	1.30	1.25	1.17	1.33
J2S3P2	1.16	1.17	1.30	1.24	1.17	1.34
J3S1P1	1.26	1.37	1.55	1.41	1.38	1.55
J3S1P2	1.30	1.43	1.61	1.44	1.41	1.60
J3S2P1	1.19	1.30	1.40	1.31	1.34	1.43
J3S2P2	1.21	1.32	1.41	1.33	1.34	1.43
J3S3P1	1.21	1.25	1.40	1.30	1.25	1.41
J3S3P2	1.20	1.24	1.35	1.28	1.24	1.37

表4 M2类算例结果

算例	算法					
	EDA1	EDA2	EDA3	EDA4	GA	SA
J1S1P1	2.63	2.63	2.63	2.63	2.63	2.63
J1S1P2	2.68	2.68	2.68	2.68	2.68	2.68
J1S2P1	1.31	1.30	1.38	1.34	1.30	1.46
J1S2P2	1.41	1.41	1.44	1.42	1.41	1.51
J1S3P1	1.41	1.41	1.43	1.41	1.40	1.54
J1S3P2	1.44	1.44	1.46	1.45	1.43	1.49
J2S1P1	1.37	1.46	1.60	1.50	1.43	1.62
J2S1P2	1.33	1.44	1.56	1.45	1.41	1.56
J2S2P1	1.20	1.25	1.36	1.29	1.26	1.40
J2S2P2	1.20	1.26	1.37	1.30	1.27	1.42
J2S3P1	1.20	1.21	1.34	1.28	1.18	1.37
J2S3P2	1.20	1.21	1.33	1.27	1.19	1.37
J3S1P1	1.31	1.45	1.58	1.44	1.43	1.58
J3S1P2	1.33	1.48	1.65	1.50	1.49	1.64
J3S2P1	1.22	1.31	1.41	1.33	1.34	1.44
J3S2P2	1.24	1.32	1.43	1.34	1.36	1.45
J3S3P1	1.23	1.28	1.42	1.33	1.25	1.43
J3S3P2	1.21	1.24	1.37	1.29	1.25	1.39

运行时间方面,4种EDA均是要长于GA、SA的。这是由于EDA生成下一代种群需要反复对概率矩阵采样生成新的个体,这一过程相对于SA中邻域搜索或GA中交叉、变异要繁杂很多,因此造成时间上的冗余。

图6给出了J2S2P1M1类典型算例的收敛趋势图。从中可以看出EDA1的收敛性更好。其他类算例的收敛趋势与此类似。

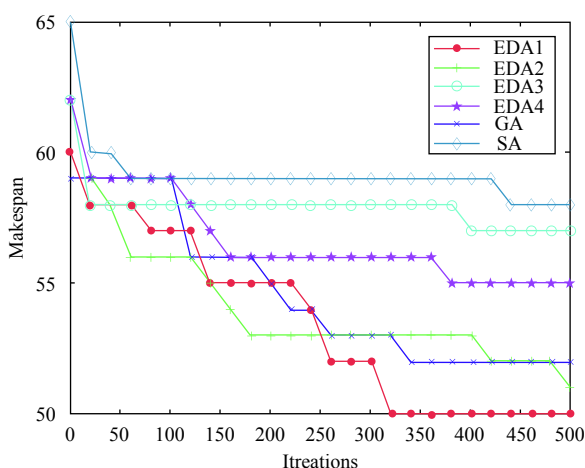


图6 J2S2P1M1类典型算例收敛趋势图

#### 4 结论与展望

本文研究了分布式估计算法在考虑差异工件的并行批处理机调度中的应用问题。首先提出了一个混合整数规划模型和一个下界,然后采用FF-LPT规则实现

对工件的分批和排序;对于基于概率矩阵的分布式估算法,提出了4种不同的更新机制,并通过仿真实验和SA、GA对比,验证了算法的有效性。经比较知,对于采取不同更新机制的EDA,EDA1的性能最好,EDA2次之,EDA3性能最差。

未来的研究可考虑通过局部优化来进一步优化算法的性能,以及考虑多阶段并行机调度问题或不同的优化目标等。

#### 参考文献

- Uzsoy R, Lee CY, Martin-Vega LA. A review of production planning and scheduling models in the semiconductor industry part II: Shop-floor control. *IIE Transactions*, 1994, 26(5): 44–55. [doi: 10.1080/07408179408966627]
- Ghazvini FJ, Dupont L. Minimizing mean flow times criteria on a single batch processing machine with non-identical jobs sizes. *International Journal of Production Economics*, 1998, 55(3): 273–280. [doi: 10.1016/S0925-5273(98)00067-X]
- Melouk S, Damodaran P, Chang PY. Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing. *International Journal of Production Economics*, 2004, 87(2): 141–147. [doi: 10.1016/S0925-5273(03)00092-6]
- 周盛超. 差异工件机器批调度若干问题研究[博士学位论文]. 合肥: 中国科学技术大学, 2016.
- Ikura Y, Gimple M. Efficient scheduling algorithms for a single batch processing machine. *Operations Research Letters*, 1986, 5(2): 61–65. [doi: 10.1016/0167-6377(86)90104-5]
- Lee CY, Uzsoy R, Martin-Vega LA. Efficient algorithms for scheduling semiconductor burn-in operations. *Operations Research*, 1992, 40(4): 764–775. [doi: 10.1287/opre.40.4.764]
- Uzsoy R. Scheduling a single batch processing machine with non-identical job sizes. *International Journal of Production Research*, 1994, 32(7): 1615–1635. [doi: 10.1080/00207549408957026]
- Zhou SC, Chen HP, Xu R, *et al.* Minimising makespan on a single batch processing machine with dynamic job arrivals and non-identical job sizes. *International Journal of Production Research*, 2014, 52(8): 2258–2274. [doi: 10.1080/00207543.2013.854937]
- Chang PY, Damodaran P, Melouk S. Minimizing makespan on parallel batch processing machines. *International Journal of Production Research*, 2004, 42(19): 4211–4220. [doi: 10.1080/00207543.2013.854937]

- 10.1080/00207540410001711863]
- 10 Damodaran P, Chang PY. Heuristics to minimize makespan of parallel batch processing machines. *The International Journal of Advanced Manufacturing Technology*, 2008, 37(9–10): 1005–1013.
  - 11 Chen HP, Du B, Huang GQ. Metaheuristics to minimise makespan on parallel batch processing machines with dynamic job arrivals. *International Journal of Computer Integrated Manufacturing*, 2010, 23(10): 942–956. [doi: [10.1080/0951192X.2010.495137](https://doi.org/10.1080/0951192X.2010.495137)]
  - 12 Xu R, Chen HP, Li XP. Makespan minimization on single batch-processing machine via ant colony optimization. *Computers & Operations Research*, 2012, 39(3): 582–593.
  - 13 Xu SB, Bean JC. A genetic algorithm for scheduling parallel non-identical batch processing machines. *Proceedings of 2007 IEEE Symposium on Computational Intelligence in Scheduling*. Honolulu, HI, USA. 2007. 143–150.
  - 14 Damodaran P, Diyadawagamage DA, Ghayeb O, *et al.* A particle swarm optimization algorithm for minimizing makespan of nonidentical parallel batch processing machines. *The International Journal of Advanced Manufacturing Technology*, 2012, 58(9–12): 1131–1140.
  - 15 Zhou SC, Liu M, Chen HP, *et al.* An effective discrete differential evolution algorithm for scheduling uniform parallel batch processing machines with non-identical capacities and arbitrary job sizes. *International Journal of Production Economics*, 2016, 179: 1–11. [doi: [10.1016/j.ijpe.2016.05.014](https://doi.org/10.1016/j.ijpe.2016.05.014)]
  - 16 Zhou SC, Li XP, Chen HP, *et al.* Minimizing makespan in a no-wait flowshop with two batch processing machines using estimation of distribution algorithm. *International Journal of Production Research*, 2016, 54(16): 4919–4937. [doi: [10.1080/00207543.2016.1140920](https://doi.org/10.1080/00207543.2016.1140920)]
  - 17 Aickelin U, Burke EK, Li J. An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering. *Journal of the Operational Research Society*, 2007, 58(12): 1574–1585. [doi: [10.1057/palgrave.jors.2602308](https://doi.org/10.1057/palgrave.jors.2602308)]
  - 18 Pinedo ML. *Scheduling: Theory, Algorithms, and Systems*. 5th ed. Cham: Springer, 2016.
  - 19 Mühlenbein H, Paaß G. From recombination of genes to the estimation of distributions I. Binary parameters. *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*. Berlin, Germany. 1996. 178–187.