

文章编号:1006-5911(2013)07-1640-08

# 蚁群算法求解混合流水车间分批调度问题

宋代立, 张 洁

(上海交通大学 计算机集成制造研究所, 上海 200240)

**摘 要:**为解决混合流水车间分批调度问题,提出一种三级递阶结构的蚁群算法。算法中,第一级蚁群算法设计了一种批量大小动态结合的柔性分批策略,完成产品的批次划分;第二级蚁群算法考虑工件在各设备的加工时间和设备可用能力,设计蚂蚁设备间的转移概率,完成工序约束下各批次的设备选择;第三级蚁群算法考虑同一设备上批次顺序相关的换批时间,设计蚂蚁批次间的转移概率,完成各设备的批次排序。通过实例仿真,分别对分批算法和混合流水车间调度算法性能进行比较分析和评价,结果表明了算法的有效性和优越性。最后从生产实际出发给出算例,验证了算法的有效性和对生产实践的指导作用。

**关键词:**蚁群算法;混合流水车间;分批调度;仿真

**中图分类号:**TP18;TP312

**文献标志码:**A

## Batch scheduling problem of hybrid flow shop based on ant colony algorithm

SONG Dai-li, ZHANG Jie

(Institute of Computer Integrated Manufacturing, Shanghai Jiaotong University, Shanghai 200240, China)

**Abstract:** To solve the problem of hybrid flow shop batch scheduling, a novel Ant Colony Optimization (ACO) algorithm with three level hierarchical structures was proposed. A flexible batching strategy was put forward in the first level of ACO to complete the batch partition of products. By considering the process time and available capacity of equipment in second level of ACO, the transition probability between ant equipments was designed, and the equipment selection of each batch under process constraint was fulfilled. Through considering the relevant batch time of batch sequence on same equipment, the transfer probabilities of ants between machines were designed in third level of ACO, and batch scheduling of equipment was completed. Performance of batch algorithm and hybrid flow shop scheduling algorithm were evaluated respectively through simulation experiment, and the results demonstrated the feasibility and effectiveness of proposed algorithm. An example from the practical production was addressed to express the guidance for production practice.

**Key words:** ant colony algorithm; hybrid flow shop; batch scheduling; simulation

## 0 引言

混合流水车间调度问题 (Hybrid Flow Shop Scheduling Problem, HFSSP) 也称柔性流水车间调度问题 (flexible flow shop scheduling problem), 是流水车间调度问题和并行机调度问题的结合。混合

流水车间分批调度问题是混合流水车间调度问题的扩展, 在考虑工件分批生产的 HFSSP 时, 不但要将工件分割为适当的多个批次, 还要为各批次选取适当的加工设备和排序方案, 从而使调度方案尽可能满足所需的性能指标。该问题比传统的 HFSSP 更加复杂, 也更接近实际生产。因此, 该问题的研究具

收稿日期: 2012-06-29; 修订日期: 2012-09-24。Received 29 June 2012; accepted 24 Sep. 2012.

基金项目: 国家 863 计划资助项目 (2012AA040907); 苏州市科技计划资助项目 (SG201131)。Foundation items: Project supported by the National High-Tech. R&D Program, China (No. 2012AA040907), and the Suzhou Scientific & Technological Plan, China (No. SG201131).

有重要的理论价值和现实意义。

目前, HFSSP 的求解思路主要有整体法(integrated approaches)和分层法(hierarchical approaches)。分层法是将 HFSSP 分为多个子问题并分别求解的一类方法。陈璐等<sup>[1]</sup>研究含有阻塞限制的多阶 HFSSP, 文中采用禁忌搜索算法优化工件的第一级排序, 并采用先到先服务和最先空闲设备分配规则为剩余各级工序指派设备, 获取调度结果; Ghola-mi 等<sup>[2]</sup>研究考虑顺序相关准备时间和随机设备故障的混合流水车间动态调度问题, 采用基于工件编码的遗传算法(Genetic Algorithm, GA)优化工件序列, 并使用 SPT(shortest process time)和 Johnson 复合调度规则指派设备解码工件序列; 王圣尧等<sup>[3]</sup>将 HFSSP 分为工件排序和设备指派两个阶段, 采用分布估计算法优化工件序列, 并使用最先空闲机器规则指派设备解码工件序列。整体法是同时求解各个子问题的方法。

在本文所研究的问题中, 工件的分批对调度结果影响极大, 因此需要着重考虑。工件的分批问题由于贴近企业的生产实际, 近年来也引起了国内外学者的研究兴趣。鞠全勇等<sup>[4]</sup>提出一种柔性分批算法, 在算法进行过程中借鉴压缩技术将相邻同类工件合并为同一子批次; 孙志竣等<sup>[5]</sup>提出一种新颖的等量分批调度遗传算法, 可使子批数量的确定和子批加工顺序的安排同时得到优化; 白俊杰等<sup>[6]</sup>提出一种基于“游标”的柔性批量分割方法, 使得批量大小可以根据机床负荷进行调整; 曾强等<sup>[7]</sup>针对并行机作业车间等量分批多目标调度优化问题, 提出基于 Witness 仿真的等量分批优化方法, 来优化加工产品的批量; 马丁等<sup>[8]</sup>引入启发式规则生成基础批, 并用遗传算法对基础批进行全局搜索, 优化分批结果。

基于以上研究现状, 本文研究一个集成批量计划和混合流水车间调度的问题。采用单个模型描述整个集成问题, 并将问题的求解过程分为产品分批、设备指派和批次排序三个相互关联的阶段, 设计蚁群算法(Ant Colony Optimization, ACO)对每个阶段进行优化, 使得所有产品的最大完工时间(makespan)最小。

## 1 问题建模

### 1.1 问题描述

加工系统有  $M$  台设备和  $X$  种产品, 每种产品

由所有同一类型的工件组成, 所有工件均按照相同的工艺路径加工, 每道工序可以在多台不同设备上加工, 加工时间随设备性能的不同而变化。同一设备上加工两个不同批次时需要一定的换批时间, 换批时间和批次的加工顺序相关。

本文的调度模型给出如下假设: ①在零时刻所有工件均可被加工; ②任一工件只有在前一道工序完成后方能进入下一道工序; ③工件的工序加工时间和加工设备相关; ④不同批次的换批时间与批次的加工顺序相关, 设备的调整时间和工件的搬运时间被考虑到换批时间中; ⑤每台设备任意时刻最多只能加工一个工件; ⑥属于不同批次工件的工序之间没有先后约束; ⑦属于同一批次的工件一旦进行加工就不能中断, 设备必须加工完该批次的全部工件后, 才能加工另一批次。

### 1.2 数学模型

为描述方便, 下面给出模型中用到的数学符号的意义:

$J_x$  表示第  $x$  种产品( $x=1, 2, \dots, X$ ),  $X$  为产品类型总数;

$J_{x,p}$  表示  $J_x$  的第  $p$  道工序( $p=1, 2, \dots, P$ ),  $P$  为总工序数;

$J_{x,p,k}$  表示  $J_{x,p}$  的第  $k$  个工件( $k=1, 2, \dots, Sum(J_x)$ ),  $Sum(J_x)$  为  $J_x$  的工件数;

$M_{x,p}$  表示可以加工  $J_{x,p}$  的设备总数;

$M_{p,m}$  表示制造系统中第  $p$  道工序的第  $m$  台设备( $m=1, 2, \dots, M_p$ ),  $M_p$  为第  $p$  道工序的所有可用设备;

$L_{x,p,n}$  表示  $J_{x,p}$  的第  $n$  个批次( $n=1, 2, \dots, N_x$ ), 其中  $N_x$  为第  $x$  种工件划分的批次总数;

$W(L_{x,p,n}, M_{p,m})$  表示 0-1 变量, 1 表示  $L_{x,p,n}$  在  $M_{p,m}$  上加工;

$Sum(L_{x,p,n})$  表示  $L_{x,p,n}$  包含的工件数量;

$ST(L_{x,p,n})$  表示  $L_{x,p,n}$  的加工开始时间;

$FT(L_{x,p,n})$  表示  $L_{x,p,n}$  的加工完成时间;

$PT(L_{x,p,n})$  表示  $L_{x,p,n}$  的加工时间;

$TU_{x,p,m}$  表示  $J_x$  的单个工件在设备  $M_{p,m}$  上的加工时间;

$Set_{x_1, x_2}$  表示从  $J_{x_1}$  切换到  $J_{x_2}$  的加工准备时间, 若  $x_2$  为起始工件, 则  $x_1=0$  且  $Set_{0, x_2}=0$ ;

$\prec M$  表示批次间的偏序关系, 如果  $L_{x_1, p, n_1}$  在制造系统上的加工顺序先于  $L_{x_2, p, n_2}$  且没有其他批次

加工,则有  $\angle M(L_{x_1,p,n_1}, L_{x_2,p,n_2}) = 1$ 。

目标函数:

$$F = \min(\max_{\substack{x=1,\dots,X \\ p=1,\dots,P \\ n=1,\dots,N_x}} (FT(L_{x,p,n}))),$$

$$x = 1, \dots, X, p = 1, \dots, P, n = 1, \dots, N_x. \quad (1)$$

$$\text{s. t.} \quad \sum_{n=1}^{N_x} \text{Sum}(L_{x,p,n}) = \text{Sum}(J_x),$$

$$x = 1, \dots, X, p = 1, \dots, P, n = 1, \dots, N_x; \quad (2)$$

$$PT(L_{x,p,n}) = TU_{x,p,m} \times \text{Sum}(L_{x,p,n}),$$

$$x = 1, \dots, X, p = 1, \dots, P, n = 1, \dots, N_x,$$

$$m = \{M_{x,p} \mid W(L_{x,p,n}, M_{p,m}) = 1\}; \quad (3)$$

$$FT(L_{x,p,n}) = \text{Set}_{x_1,x} + ST(L_{x,p,n}) + PT(L_{x,p,n}),$$

$$x = 1, \dots, X, p = 1, \dots, P, n = 1, \dots, N_x,$$

$$x_1 = \{J_x \mid \angle M(L_{x_1,p,n_1}, L_{x,p,n}) = 1\},$$

$$n_1 = 1, \dots, N_{x_1}; \quad (4)$$

$$ST(L_{x,p,n}) \geq \max(FT(L_{x_1,p,n_1}), FT(L_{x,p-1,n})),$$

$$x = 1, \dots, X, p = 1, \dots, P, n = 1, \dots, N_x,$$

$$x_1 = \{J_x \mid \angle M(L_{x_1,p,n_1}, L_{x,p,n}) = 1\},$$

$$n_1 = 1, \dots, N_{x_1}; \quad (5)$$

$$\sum_{m \in M_{x,p}} W(L_{x,p,n}, M_{p,m}) = 1,$$

$$x = 1, \dots, X, p = 1, \dots, P, n = 1, \dots, N_x; \quad (6)$$

$$\sum_{n_1 \in \Phi(J_{n_1})} \text{Sum}(L_{x,p,n_1}) \geq \sum_{n_1 \in \Phi(J_{n_2})} \text{Sum}(L_{x,p+1,n_2}),$$

$$x = 1, \dots, X, p = 1, \dots, P-1,$$

$$\Phi(J_{n_1}) = \{N_x \mid FT(L_{x,p,n}) \leq T\},$$

$$\Phi(J_{n_2}) = \{N_x \mid FT(L_{x,p+1,n}) \leq T\},$$

$$n = 1, \dots, N_x. \quad (7)$$

其中:式(1)表示调度目标为最小化最大完工时间;式(2)表示加工数量约束,即对每种产品进行分批,各批次包含的工件的数量之和等于产品包含的工件数量;式(3)表示批次的加工时间等于单个工件的加工时间和批次包含的工件数量之积;式(4)表示批次的完成时间等于批次的换批时间、加工时间和开始加工时间之和;式(5)表示批次的开始加工时间不早于同一设备上的前一批次和同一批次的前道工序完成时间的最大值;式(6)表示设备占用约束,即同一批次的所有工件只能在一台设备上加工;式(7)表示批次约束关系,若某时刻  $J_{x,p}$  的所有完成批次数量和为  $Q$ ,则  $J_{x,p+1}$  在该时刻所有完成批次的数量和应不大于  $Q$ 。

### 1.3 分批策略

在车间的分批调度问题中,批量大小和生产周

期存在 U 型关系<sup>[9]</sup>,批量过大或过小都不利于提高车间的运行效率。当批量过小时,批次的数量增加,问题的搜索空间也会相应增大,影响搜索结果,且批次数量的增大会导致换批频繁;当批量过大时,较大的批量占有当前设备,会使后续设备处于闲置等待状态。

适当的批量分割方法能在不明显影响搜索效率的情况下有效减少设备的空闲等待时间,提高生产效率,缩短周期。孙志竣等<sup>[5]</sup>采用等量分批策略,但由于不同设备的性能不同,等量分批无法根据设备性能调整批量大小,会造成设备负载不均衡,不利于缩短生产周期。鞠全勇等<sup>[4]</sup>借鉴压缩技术设计了一种批量大小和排序动态结合的策略,但算法以每个工件作为一个批次,随着产品所包含的工件数目的增加,搜索空间会急剧增加,求解时间和求解效率会下降。针对以上分批策略的不足,本文借鉴批量大小动态结合的思想,提出一种柔性批量分割方法。批量分割时,将同一产品根据其包含的工件数量划分为多个任务,每个任务包含若干个工件,以每个任务作为一个子批量。在排序过程中,把相邻的属于同一产品的任务合并为一个批次。例如:三类产品 A,B,C,每类产品各含 20 个工件,假定以 5 个工件为一个任务,属于各产品的任务分别记为  $\bar{A}$ ,  $\bar{B}$  和  $\bar{C}$ ,若初始排序为  $\bar{B}\bar{B}\bar{A}\bar{A}\bar{B}\bar{B}\bar{C}\bar{A}\bar{A}\bar{C}\bar{C}\bar{C}$ ,其批次数量为 12,经过动态结合后的排序为  $\bar{B}^2 \bar{A}^2 \bar{B}^2 \bar{C}\bar{A}^2 \bar{C}^3$ ,批次数量为 6。

## 2 混合流水车间分批调度的蚁群算法

蚁群算法模拟蚁群寻找通往食物源最短路径的信息交换机制,被广泛用来解决组合优化问题。图 1 所示为本文的调度过程图,参照分层法将分批调度过程分为产品分批、设备指派和批次排序三个阶段,并设计蚁群算法进行优化。算法采用三层嵌套结构运行:第一级(产品分批层)蚁群算法为产生分批方案的分批算法,第二级(设备指派层)和第三级(批次排序层)蚁群算法组成混合流水车间调度算法,它以分批的结果为调度对象,通过两层嵌套的形式搜索当前分批结果的最优排产方案。

假设有 2 种产品,每种产品包含 4 个工件且需要经历 2 道加工工序,每道加工工序有 2 台设备可以加工,记为  $\{1, 2, 3, 4\}$ 。选定 2 个工件为一个任务,则所有产品可分为 4 个任务,分别为  $\{1_1, 1_2, 2_1, 2_2\}$ ,其中“ $1_2$ ”表示产品 1 的第 2 个任务。若经过产

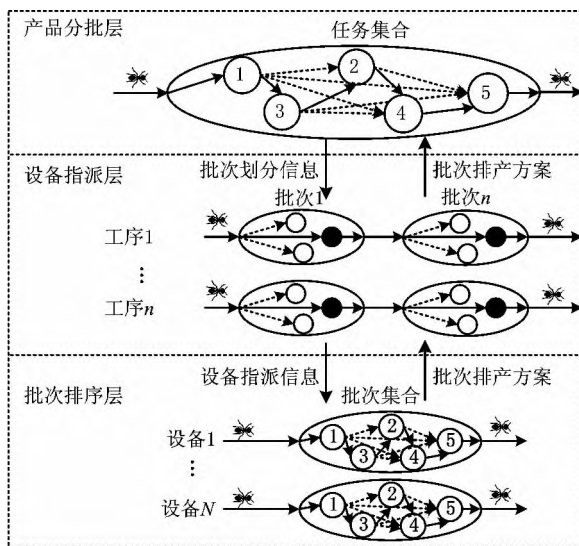


图1 调度过程图

品分批层蚁群算法搜索到的序列为 $\{1_1, 2_1, 2_2, 1_2\}$ , 合并“ $2_1$ ”和“ $2_2$ ”得到的分批方案为 $\{1_1^2, 2_1^4, 1_2^2\}$ , 其中“ $2_1^4$ ”表示产品2的第1个批次, 工件数量为4; 设备指派层蚁群算法以分批结果为输入, 搜索批次的设备指派方案, 若搜索得到 $\{1_1^2, 2_1^4, 1_2^2\}$ 三个批次的设备指派序列为 $\{1, 2, 1, 3, 4, 4\}$ (左边表示工序1的设备指派方案, 右边表示工序2的设备指派方案), 则可以确定各设备的加工批次为 $\{1: 1_1^2, 1_2^2; 2: 2_1^4; 3: 1_1^2; 4: 2_1^4, 1_2^2\}$ ; 批次排序层蚁群算法以各设备的加工批次结果为输入搜索设备上的批次排序方案, 若搜索到的批次排序序列为 $\{2, 1; 1; 1; 1, 2\}$ (用分号区分不同设备的批次排序方案), 则可以确定各设备的批次加工方案为 $\{1: 1_2^2, 1_1^2; 2: 2_1^4; 3: 1_1^2; 4: 2_1^4, 1_2^2\}$ , 其排产甘特图如图2所示(黑色表示换批时间)。

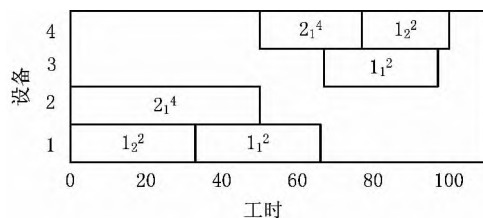


图2 蚁群算法排产甘特图

## 2.1 算法过程

算法运行前设定的参数包括: 蚁群算法中的蚂蚁数 $Q^1, Q^2$ 和 $Q^3$ ; 信息素挥发系数 $\rho_1, \rho_2$ 和 $\rho_3$ ; 循环计数器 $q$ , 蚂蚁个数计数器 $r_1, r_2$ 和 $r_3$ , 循环次数 $Q$ ; 最优路径 $C_{\max}^{\text{best}}$ 等。之后进入如下步骤:

### 步骤1 生成初始任务序列。

根据产品 $J_x (x=1, 2, \dots, X)$ 中任务包含的工件数将其划分为若干任务 $R_i (i=1, 2, \dots, N)$ , 生成一只蚂蚁 $a_1$ , 并随机选定一个任务(如 $R_i$ )作为首个游历的节点。设已游历任务计数器 $s_1=1$ , 蚂蚁 $a_1$ 的第 $s_1$ 步可以游历的任务集合为 $W^{a_1}(s_1)=\{R_1, R_2, \dots, R_N\}-\text{tabu}^{a_1}(s_1)$ , 其中 $\text{tabu}^{a_1}(s_1)$ 表示蚂蚁 $a_1$ 第 $s_1$ 步已游历的工件集合。蚂蚁 $a_1$ 在第 $s_1$ 步根据状态转移规则从 $W^{a_1}(s_1)$ 中选取节点, 并使 $s_1=s_1+1$ 。重复节点选取过程直到 $s_1=N$ , 即蚂蚁遍历所有任务节点, 得到任务序列。设蚂蚁个数计数器 $r_1=r_1+1$ 。

### 步骤2 生成批次序列并对批次进行调度。

批次序列的生产过程参考分批策略, 批次调度分为设备指派和批次排序两个阶段, 过程如下:

**步骤2.1 指派初始工序设备。**初始化各设备的可用能力 $C_k (k=1, \dots, M)$ , 为工序 $p (p=1, \dots, P)$ 生成一只蚂蚁 $a_2(p)$ 。设已游历批次计数器 $s_2=0$ , 蚂蚁第 $s_2$ 步从设备集合为 $M_{x,p}$ 中选取设备, 并使 $s_2=s_2+1$ , 更新设备能力信息 $C_k=C_k-PT(L_{x,p,n})$ ,  $L_{x,p,n}$ 表示第 $s_2$ 步对应的批次节点,  $W(L_{x,p,n}, M_{p,k})=1$ 。重复设备选取过程直到 $s_2=\sum L_{x,p,n}$ , 即蚂蚁遍历所有批次节点, 得到最终的设备指派方案。设蚂蚁个数计数器 $r_2=r_2+1$ 。

**步骤2.2 对各设备进行批次排序。**根据步骤2.1的设备指派方案, 统计各设备需要加工的批次。批次排序过程如下:

**步骤2.2.1 生成设备的初始批次序列。**初始化工序计数器 $p$ (令 $p=1$ ), 为设备集 $M_p$ 中每一台设备 $m$ 生成一只蚂蚁 $a_3(m)$ , 并根据设备 $m$ 上批次转移规则确定批次序列(过程参考步骤1)。

**步骤2.2.2 序列解码, 得到蚂蚁 $a_3(m)$ 游历所得序列的调度结果。**解码过程如下:

(1) 若为初始工序( $p=1$ ), 则初始化 $C_{\max 3}$ , 令 $C_{\max 3}=0$ , 否则转(2)。

(2) 从设备集 $M_p$ 中取出设备 $k$ , 由式(3)~式(5)计算设备 $k$ 上所有批次( $L_{x,p,n} \in \{W(L_{x,p,n}, M_{p,k})=1\}$ )的加工开始时间 $ST(L_{x,p,n})$ 和加工完成时间 $FT(L_{x,p,n})$ , 更新设备集 $M_p=M_p \setminus k$ 。

(3) 若 $C_{\max 3} < FT(L_{x,p,n})$ , 则更新 $C_{\max 3}$ , 即 $C_{\max 3}=FT(L_{x,p,n})$ , 转(4), 否则直接转(4);

(4) 若 $M_p=\Phi$ , 则转入步骤2.2.3, 否则返回

(2)。

步骤 2 2 3 判断  $p < P$  是否成立, 成立则  $p = p + 1$ , 转步骤 2 2 1; 否则判断  $r_3 < Q^3$  是否成立, 成立则  $p = 1$ , 转入步骤 2 2 1, 否则转下一步。

步骤 2 2 4 若连续 5 次搜索结果  $C_{\max}^{\text{local3}}$  无变化, 则转步骤 2 3; 否则转下一步。

步骤 2 2 5 更新蚁群算法的信息素浓度。

若  $C_{\max}^{\text{best}} < C_{\max}^{\text{local3}}$ , 则  $C_{\max}^{\text{best}} = C_{\max}^{\text{local3}}$ , 更新最优调度方案。信息素更新的方法为: 首先对信息素作挥发处理, 公式为  $\tau_{n,i}(q+1) = \rho \cdot \tau_{n,i}(q)$ 。再对本次循环中取得最短完工时间的那只蚂蚁游历的路径增加信息素, 公式为  $\tau_{n,i}^{a_3(\min)} = \tau_{n,i}^{a_3(\min)} + \Delta\tau_{n,i}^{\text{best}}$ , 其中  $a_3(\min)$  为取得最短路径的蚂蚁,  $\Delta\tau_{n,i}^{\text{best}} = 1/C_{\max}^{\text{best}}$ 。转步骤 2 2 1。

步骤 2 3 获取搜索到的调度结果。蚂蚁  $a_2(p)$  搜索到的路径为  $C_{\max2} = C_{\max}^{\text{local3}}$ , 并更新调度方案。若  $r_2 < Q^2$ , 则转步骤 2 1, 否则转下一步。

步骤 2 4 若连续 5 次搜索结果  $C_{\max}^{\text{local2}}$  无变化, 则转步骤 3, 否则转下一步。

步骤 2 5 更新信息素浓度。若  $C_{\max}^{\text{best}} < C_{\max}^{\text{local2}}$ , 则  $C_{\max}^{\text{best}} = C_{\max}^{\text{local2}}$ , 更新最优调度方案。信息素更新的方法参考 2 2 5 节。转步骤 2 1。

步骤 3 获取搜索到的调度结果。

蚂蚁  $a_1$  搜索到的路径为  $C_{\max1} = C_{\max}^{\text{local2}}$ , 并更新调度方案。判断  $r_1 < Q^1$  是否成立, 是则转步骤 1, 否则转下一步。

步骤 4 更新搜索到的局部最优值  $C_{\max}^{\text{local1}}$ 。若  $q > Q$ , 则算法结束, 否则转下一步。

步骤 5 更新信息素浓度。

若  $C_{\max}^{\text{best}} < C_{\max}^{\text{local1}}$ , 则  $C_{\max}^{\text{best}} = C_{\max}^{\text{local1}}$ , 更新最优调度方案, 信息素更新的方法参考 2 2 5 节。转步骤 1。

## 2 2 蚁群状态转移规则

在产品分批阶段, 蚂蚁在构建解的过程中采用伪随机比例的状态转移规则<sup>[10]</sup>来选择下一步要加工的任务:  $q_1$  为初始时刻设定的参数, 且  $0 \leq q_1 \leq 1$ ;  $q$  为一个随机数,  $q \in [0, 1]$ 。若  $q \leq q_1$ , 则根据  $\max_{i \in W^{a_1}(n)} \{(\tau_{n,i}^{I,a_1})^\alpha \cdot (\eta_{n,i}^{I,a_1})^\beta\}$  选择下一个节点; 若  $q_1 > q$ , 则按照式(8)计算节点选择概率。

$$P_{n,i}^{I,a_1} = \frac{(\tau_{n,i}^{I,a_1})^\alpha \cdot (\eta_{n,i}^{I,a_1})^\beta}{\sum_{i \in W^{a_1}(n)} (\tau_{n,i}^{I,a_1})^\alpha \cdot (\eta_{n,i}^{I,a_1})^\beta} \quad (8)$$

式中:  $\tau_{n,i}^{I,a_1}$  为任务  $(n, i)$  间的信息素水平;  $\eta_{n,i}^{I,a_1} = 1/((\sum_{p=1}^P \text{Set}_{J(n), J(i)} + \max_{p \in P} (PT(R_i), PT(R_n))))$ 。

在设备指派阶段, 每个蚂蚁  $a_2(p)$  的状态转移概率为

$$P_{k(n), k(i)}^{II, a_2(p)} = \frac{(\tau_{k(n), k(i)}^{II, a_2(p)})^\alpha \cdot (\eta_{k(n), k(i)}^{II, a_2(p)})^\beta}{\sum_{i \in M_p} (\tau_{k(n), k(i)}^{II, a_2(p)})^\alpha \cdot (\eta_{k(n), k(i)}^{II, a_2(p)})^\beta} \quad (9)$$

式中:  $\tau_{k(n), k(i)}^{II, a_2(p)}$  为批次  $(n, i)$  所选设备的信息素水平;  $\eta_{k(n), k(i)}^{II, a_2(p)} = C_k / PT(L_{x,p,i})$ ,  $L_{x,p,i}$  表示批次节点  $i$ 。

在批次排序阶段, 每个蚂蚁  $a_3(m)$  的状态转移概率为

$$P_{n,i}^{III, a_3(m)} = \frac{(\tau_{n,i}^{III, a_3(m)})^\alpha \cdot (\eta_{n,i}^{III, a_3(m)})^\beta}{\sum_{W^{a_3(m)}(n)} (\tau_{n,i}^{III, a_3(m)})^\alpha \cdot (\eta_{n,i}^{III, a_3(m)})^\beta} \quad (10)$$

式中:  $\tau_{n,i}^{III, a_3(m)}$  为批次  $(n, i)$  间的信息素水平;  $\eta_{n,i}^{III, a_3(m)} = 1/(\text{Set}_{J(n), J(i)} + FT(L_{J(i), p-1, i}) + PT(L_{J(i), p, i}))$ ,  $J(n)$  和  $J(i)$  分批表示批次  $n$  和  $i$  的产品种类。

## 3 实验分析

鉴于国内外尚无相同的混合流水车间分批调度问题的标准算例, 为验证本文算法的性能, 参照文献[5]设计针对混合流水车间的算例, 验证分批算法的优越性, 而后用文献[2]中的算例验证混合流水车间调度算法的优越性, 最后针对企业的实际数据进行仿真实验。算法运行环境为 Core2 E8400 CPU(2.8 GHz), 2 G 内存, Windows XP 操作系统, 并采用 C# 语言编程。

### 3.1 仿真试验 1

表 1 所示为仿真试验的测试算例, 对表中仿真试验 1 的算例, 以 2 个工件为一个任务, 分别采用蚁群算法和文献[11]提出的遗传算法进行验算, 用本文第二级和第三级的蚁群算法替代文献[11]中的模拟退火(Simulated Annealing, SA)算法, 求解批次的调度方案。图 3 和图 4 所示为两种分批算法的甘特图, 图中数字代表产品种类和子批批次, 如“3, 2”表示第 3 种产品的第 2 个批次, 黑色部分表示换批时间。蚁群算法将所有产品分为 16 个批次, 分别为 4(2, 2, 2, 2), 4(2, 2, 2, 2), 4(2, 2, 2, 2) 和 4(2, 2, 2, 2), 所得生产周期为 413; 遗传算法将所有产品分为 16 个批次, 分别为 4(1, 3, 2, 2), 4(2, 2, 2, 2), 4(1, 3, 3, 1) 和 4(3, 2, 2, 1), 所得生产周期为 492。从图中

可以看出,蚁群算法能根据设备的加工能力合理安排批量的大小,求出的结果在设备利用率和设备的负载均衡方面均优于遗传算法,即在相同的批次数下,本文提出的蚁群算法更能提高设备利用率,缩短生产周期。

表 1 仿真试验测试算例

参数	仿真试验 1	仿真试验 2
产品种类/工件数	4	6,30,100
工序数	4	2,4,8
每道工序并行机数量	U[2,4]	U[1,5]
设备加工时间	U[20,30]	U[20,30]
换批时间(不同产品)	U[5,7.5]	U[12,24]
换批时间(同种产品)	U[1,3]	U[3,5]
产品包含的工件数	8	—

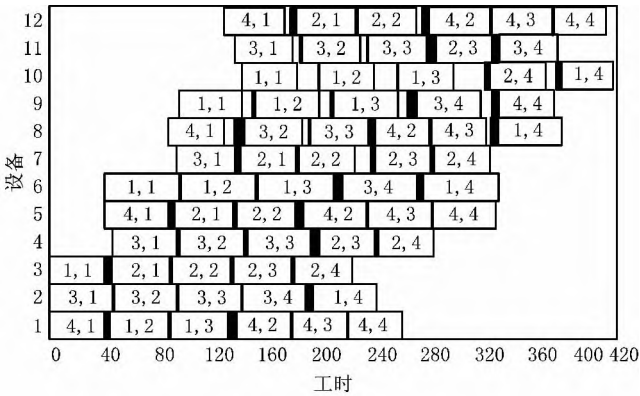


图3 蚁群算法甘特图

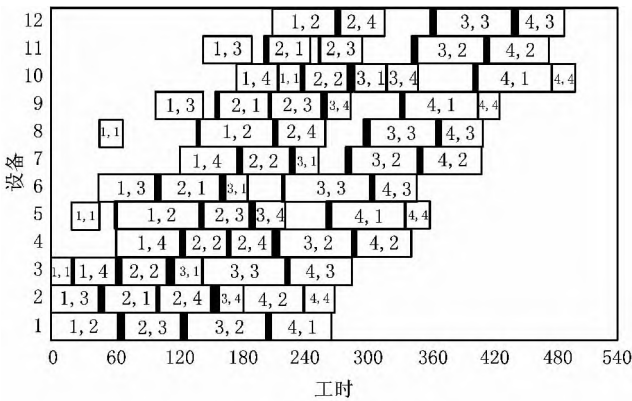


图4 模拟退火算法甘特图

3.2 仿真试验 2

为验证混合流水车间调度算法的优越性,采用参考文献[2]给出的算例进行验算。仿真试验测试算例如表 1 所示,对表中仿真试验 2 的算例用蚁群算法(ACO)、遗传算法<sup>[2]</sup>(GA)和模拟退火算法<sup>[12]</sup>(SA)分别运行 10 次,得到计算时间的平均值及计算结果的平均值(avg)和最优值(min)。求解结果如表 2 所示。表中实验组以工件数和工序数表示,如“6×2”表示 6 个工件,每个工件要经过 2 道工序。分析可知,对于所有算例,ACO 在计算时间上均优于其他两种算法,且计算时间的优势随着问题规模的增大越来越明显。从计算结果来看,ACO 对所有算例的求解结果的平均值和最优值均优于其他两种算法,SA 的求解结果优于 GA。对比 ACO 和次优的 SA,定义 ACO 相对于 SA 的改进率  $\delta = (SA_{avg} - ACO_{avg}) / ACO_{avg}$ ,描述 ACO 相对于 SA 求解结果的改进程度,工件数量分别为 6(6×2, 6×4, 6×

表 2 算法对比分析

实验组	最大完工时间/min						计算时间/ms		
	ACO		GA		SA		ACO	GA	SA
	最小值	平均值	最小值	平均值	最小值	平均值			
6×2	249	249.8	249	250.4	249	250.2	223.5	348.6	236.7
30×2	1 138	1 160.8	1 152	1 181.7	1 143	1 170.5	2 232.7	2 984.2	2 346.7
100×2	3 786	3 818.6	3 812	3 870.5	3 792	3 845.6	18 217.2	20 365.3	19 231.5
6×4	589	590.6	589	592.3	589	593.6	578.6	732.1	657.8
30×4	2 306	2 341.2	2 321	2 362.6	2 312	2 354.6	8 032.5	9 536.8	8 973.2
100×4	7 769	7 815.6	7 786	7 882.5	7 771	7 891.7	80 326.5	88 567.3	82 364.5
6×8	835	838.6	841	845.2	839	843.1	1 752.6	1 936.8	1 854.2
30×8	2 705	2 748.6	2 712	2 776.5	2 716	2 773.9	32 461.7	34 136.2	33 286.4
100×8	7 982	8 040.6	8 026	8 173.6	7 995	8 156.9	372 136.5	393 642.8	386 284.6

8),  $30(30 \times 2, 30 \times 4, 30 \times 8)$  和  $100(100 \times 2, 100 \times 4, 100 \times 8)$  时, 对于不同工序数, ACO 相对于 SA 的改进率平均值分别为 0.4%, 0.78% 和 1.04%; 同理, 当工序数量分别为  $2(6 \times 2, 30 \times 2, 100 \times 2)$ ,  $4(6 \times 4, 30 \times 4, 100 \times 4)$  和  $8(6 \times 8, 30 \times 8, 100 \times 8)$  时, 不同工件数 ACO 相对于 SA 求解结果的平均改进程度分别为 0.57%, 0.68% 和 0.97%。可见随着问题规模的增加, 蚁群算法的优化效果越来越明显。

### 3.3 实例仿真

某印刷电路板 (Printed Circuit Board, PCB) 装配车间有 4 个加工区间 (表面贴装技术 (Surface Mounted Technology, SMT) 贴片加工区、插件加工区、补焊加工区和测试区) 共 9 条生产线, 分别为  $3(S_1, S_2, S_3)$ ,  $2(M_1, M_2)$ ,  $2(A_1, A_2)$  和  $2(T_1, T_2)$ 。某一时段加工 5 种 PCB 板, 每种 PCB 板的数量均为 1 000。每个 PCB 均要经过这 4 个加工区。PCB 的加工时间和加工准备时间如表 3 和表 4 所示。

表 3 各条生产线的加工时间

PCB	$S_1$	$S_2$	$S_3$	$M_1$	$M_2$	$A_1$	$A_2$	$T_1$	$T_2$
1	0.923	0.923	0.882	0.60	1.50	0.60	0.40	0.60	0.40
2	1.428	0.923	1.200	0.40	0.30	0.60	0.30	0.60	0.55
3	1	1.428	1	0.40	0.24	0.60	0.30	0.55	0.50
4	1	1	1	0.06	0.24	0.75	0.60	0.55	0.50
5	1	1	0.600	0.06	1.50	0.75	0.75	0.40	0.55

表 4 不同批次 PCB 的换批时间

PCB	1	2	3	4	5
1	1.52	7.85	13.34	13.01	12.52
2	5.03	1.38	10.12	5.85	14.34
3	8.02	9.95	1.28	5.97	11.62
4	10.12	10.88	10.79	1.56	10.36
5	6.54	12.98	6.34	14.63	1.23

分别采用本文的 ACO、文献[11]的遗传—模拟退火算法 (GA-SA), 以及由本文第二级和第三级蚁群算法组成的不分批的蚁群算法 (Ant Colony Algorithm Without Lot-Streaming, ACOWLS), 对以上实际算例进行求解, 求解结果如表 5 所示。分析可知, 采用分批调度的方法可以明显缩短生产周期, 提高设备利用率。ACO 和 GA-SA 均以 100 个 PCB 为最小的分批单元, ACO 将 PCB 分为 {10, 10, 9, 7, 7} 共 43 个批次, GA-SA 将 PCB 分为 {10, 7, 9, 6, 9} 共 41 个批次, 可见在未明显提高批次数量的情

况下, 蚁群算法能搜索到更优的结果。

表 5 算法求解结果对比

算法	求解结果	批次数量	平均设备利用率/%
ACO	1 542.68	43	93.5
GA-SA	2 153.46	41	87.6
ACOWLS	2 991.37	5	63.2

## 4 结束语

本文针对混合流水车间分批调度问题, 建立了集成分批和调度的数学模型, 针对分批和调度两个阶段对目标值的影响, 将分批调度过程分为产品分批、设备指派和批次排序三个阶段, 并设计了一种三层递阶结构的蚁群算法, 分别对各阶段进行优化求解。在产品分批的蚁群算法中借鉴压缩技术, 设计了一种批量大小动态结合的柔性分批策略, 使得产品能够根据设备负载柔性调整批量大小。通过实例仿真, 分别对分批算法和混合流水车间调度算法的性能进行比较分析和评价, 结果表明了算法的有效性和优越性。最后给出企业的实际算例, 其结果可行, 对生产过程具有一定的指导作用。

## 参考文献:

- [1] CHEN Lu, XI Lifeng, CAI Jianguo, et al. A tabu search algorithm for hybrid flow shop problem with blocking constraint [J]. Journal of Shanghai Jiaotong University, 2006, 40(5): 856-859 (in Chinese). [陈璐, 奚立峰, 蔡建国, 等. 一种求解带有阻塞限制的混合流水车间的紧急搜索算法 [J]. 上海交通大学学报, 2006, 40(5): 856-859.]
- [2] GHOLAMI M, ZANDIEH M, TABRIZ A A. Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns [J]. International Journal of Advanced Manufacturing Technology, 2009, 42(1/2): 189-201.
- [3] WANG Shengyao, WANG Ling, XU Ye, et al. An estimation of distribution algorithm for solving hybrid flow-shop scheduling problem [J]. Acta Automatica Sinica, 2012, 38(3): 437-443 (in Chinese). [王圣尧, 王凌, 许烨, 等. 求解混合流水车间调度问题的分布估计算法 [J]. 自动化学报, 2012, 38(3): 437-443.]
- [4] JU Quanyong, ZHU Jianying. Multi-objective flexible job shop scheduling of batch production [J]. Chinese Journal of Mechanical Engineering, 2007, 43(8): 148-154 (in Chinese). [鞠全勇, 朱剑英. 多目标批量生产柔性作业车间优化调度 [J]. 机械工程学报, 2007, 43(8): 148-154.]
- [5] SUN Zhijun, AN Jin, HUANG Weiqing. Lot scheduling with

- multiple process routes in Job Shop[J]. China Mechanical Engineering, 2008, 19(2): 183-187 (in Chinese). [孙志竣, 安进, 黄卫清. 作业车间多工艺路线批量作业计划优化[J]. 中国机械工程, 2008, 19(2): 183-187.]
- [6] BAI Junjie, GONG Yiguang, WANG Ningsheng, et al. Multi-objective flexible Job Shop scheduling with lot-splitting[J]. Computer Integrated Manufacturing Systems, 2010, 16(2): 396-403 (in Chinese). [白俊杰, 龚毅光, 王宁生, 等. 多目标柔性作业车间分批优化调度[J]. 计算机集成制造系统, 2010, 16(2): 396-403.]
- [7] ZENG Qiang, YANG Yu, WANG Xiaolei, et al. Multi-objective optimization method for equal lot scheduling problem of Job Shop with parallel machines[J]. Computer Integrated Manufacturing Systems, 2011, 17(4): 816-825 (in Chinese). [曾强, 杨育, 王小磊, 等. 并行机作业车间等量分批多目标优化调度[J]. 计算机集成制造系统, 2011, 17(4): 816-825.]
- [8] MA Ding, CHEN Qingxin, MAO Ning, et al. Parallel machine batch scheduling for due date constraints and setup time[J]. Computer Integrated Manufacturing Systems, 2012, 18(1): 111-117 (in Chinese). [马丁, 陈庆新, 毛宁, 等. 具有交货期约束带准备时间的平行机分批调度[J]. 计算机集成制造系统, 2012, 18(1): 111-117.]
- [9] KARMARKAR U S, KEKRE S. Lot-sizing in multi-item multi-machine job shops[J]. IIE Transactions, 1985, 17(3): 290-298.
- [10] GAMBARDELLA L M, DORIGO M. Solving symmetric and asymmetric TSPs by ant colonies[C]// Processings of IEEE International Conference on Evolutionary Computation. Piscataway, N. J., USA: IEEE Press, 1996: 622-627.
- [11] HUANG Gang, PANG Zhiguo, HU Zhenhua. Algorithms for lot split scheduling with setup time[J]. Industrial Engineering Journal, 2007, 10(6): 91-95 (in Chinese). [黄刚, 庞志国, 胡振华. 带加工准备时间的作业车间分批调度算法[J]. 工业工程, 2007, 10(6): 91-95.]
- [12] LOW C. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines[J]. Computer and Operations Research, 2005, 32(8): 2013-2025.

#### 作者简介:

宋代立(1987—),男,湖北武汉人,硕士研究生,研究方向:生产调度和智能算法,E-mail:sdl8724@163.com;

张洁(1963—),女,江苏江阴人,教授,博士生导师,研究方向:制造信息工程、智能制造系统及制造型服务等。