

DOI: 10.3901/JME.2014.24.191

## 存在多设备工序的综合调度算法\*

谢志强<sup>1,2</sup> 齐永红<sup>1</sup> 杨 静<sup>2</sup>

(1. 哈尔滨理工大学计算机学院 哈尔滨 150080;

2. 哈尔滨工程大学计算机学院 哈尔滨 150001)

**摘要:** 针对工序集中存在需要多台设备协同完成加工的多设备工序的综合调度问题, 提出一种把多设备工序设计成在多个相关设备上同时加工的多个虚拟工序的方法。该算法把产品加工树转化为多设备工序被设计成虚拟工序组的虚拟加工工艺树, 为了优先调度虚拟工序组, 将虚拟加工工艺树分解为以虚拟工序组为根结点的子树和剩余标准工序组成的子树, 优先调度以虚拟工序组为根结点的子树; 当以虚拟工序组为根结点的子树不唯一时, 按子树上虚拟工序组个数多少确定子树的调度次序; 对每个以虚拟工序组为根结点的子树, 优先调度虚拟工序组个数较多的路径上工序; 为了使相关虚拟工序同时开始加工, 提出虚拟工序动态调整策略。理论分析和实例表明, 所提出的调度算法能够较好地解决存在多设备工序的综合调度问题。

**关键词:** 综合调度; 多设备工序; 虚拟工序; 虚拟工序组; 虚拟工序动态调整策略

**中图分类号:** TP278

## Integrated Scheduling Algorithm with Multiple-devices-operation

XIE Zhiqiang<sup>1,2</sup> QI Yonghong<sup>1</sup> YANG Jing<sup>2</sup>

(1. College of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080;

2. College of Computer Science and Technology, Harbin Engineering University, Harbin 150001)

**Abstract:** Aiming at integrated scheduling problem with multiple-devices-operation, a method to change multiple-devices-operation into multiple related virtual operations which processes on multiple related machines at the same time is proposed. Product processing tree is changed into virtual processing tree, on which each multiple-devices-operation is changed into a virtual operation group. Virtual processing tree is decomposed into subtree whose root node is virtual operation group and subtree that composed of remaining standard operations. The subtree whose root node is virtual operation group is scheduled previously; When the subtree whose root node is virtual operation group is not unique, the scheduling order of the subtree is determined according to the number of virtual operation group. The scheduling order of operations on subtree whose root node is virtual operation group is determined according to the number of virtual operation group on the path. In order to make related virtual operations start at the same time, virtual operation dynamic adjustment strategy is proposed. Theoretical analysis and examples show that the scheduling algorithm proposed is an effective solution to the scheduling problem with multiple-devices-operation.

**Key words:** integrated scheduling; multiple-devices-operation; virtual operation; virtual operation group; virtual operation dynamic adjustment strategy

### 0 前言

传统的产品生产制造过程是将产品按工件分解, 采取先加工后装配的方式制造, 对大批量相同产品采取流水调度(Flow-shop), 对多品种小批量产

品采取车间调度(Job-shop), 因此相应的纯加工调度和纯装配调度得到了深入的研究, 如遗传算法<sup>[1-3]</sup>、模拟退火法<sup>[4]</sup>、禁忌搜索法<sup>[5]</sup>、神经网络法<sup>[6]</sup>、启发式算法<sup>[7]</sup>及各种混合算法<sup>[8-11]</sup>。

随着社会对产品个性化需求的增加, 多品种、小批量的生产模式越来越受关注。对于多品种小批量产品, 特别是树状结构的复杂单产品, 如果采取先加工后装配的方式制造, 必然割裂产品加工和装配内在可并行处理的关系, 影响产品制造效率。于是, 提出了加工和装配一同处理的复杂单产品综合

\* 国家自然科学基金(60873019, 61370086, 61370083)、黑龙江省自然科学基金(F201101)、黑龙江省教育厅科技(12531105)、黑龙江省博士后科研启动(LBH-Q13092)和现代制造技术与刀具开发黑龙江省高校重点实验室开放课题(2012DJ-003)资助项目。20131222 收到初稿, 20140617 收到修改稿

调度算法。目前,复杂单产品综合调度算法已取得一些成果,如拟关键路径算法(Allied critical path method, ACPM)<sup>[12]</sup>、解决设备驱动的调度算法<sup>[13]</sup>、存在调整时间综合调度工序的算法<sup>[14]</sup>及解决存在设备无关延迟约束的调度算法<sup>[15]</sup>等。

在装配制造企业的实际生产中还存在单个工序需要多个设备共同加工的情况,例如,当焊接两个大型零件的时候需要两个设备分别来固定它们,同时还需要一个焊接设备进行焊接,此时该焊接工序同时需要三个设备来共同完成。目前尚鲜见相关研究成果的报道,因此有必要解决单件复杂产品存在多设备共同完成单个特殊工序的综合调度问题。

为了解决工序集中存在多设备工序的综合调度问题,本文借鉴文献[16],将该问题的工艺树中的多设备工序设计为包含多个相关虚拟工序的虚拟工序组,形成特殊的虚拟加工工艺树;设计相应的工序调度次序方案,使虚拟工序组中相关虚拟工序在相应的设备上尽早同时并行加工。由于虚拟工序组中虚拟工序对于加工设备的要求较高,为了让虚拟工序组少受干扰,本文设计了优先调度虚拟工序组中虚拟工序的策略,并通过实例比较表明了所提算法的优越性。

## 1 问题描述及分析

### 1.1 问题描述

复杂产品的加工工艺树是根据产品工序之间的制约关系建立的树状结构,加工工艺树上的结点代表产品的工序,有向边代表工序加工次序的偏序关系,根结点代表产品的最后一个工序,根结点工序加工完毕时,表明该产品加工完毕。当工序集中存在多设备工序时,复杂单产品调度问题可以描述为必须满足以下 4 个约束条件时,使得产品总加工时间最短。

(1) 每个工序必须等待其所有的工艺紧前和设备紧前工序加工结束后才能进行加工。

(2) 每个设备任意时刻只能加工一个工序,且加工过程不能中断。

(3) 设备集中不存在具有相同功能的设备。

(4) 工序集中允许存在多个相关设备共同加工的多设备工序。

设有  $n$  个工序的产品  $P$  在  $m$  个设备上加工,由于本文研究的复杂单产品综合调度问题是设计一种合适的调度算法,使每个工序尽早开始加工,最终

实现产品最后完工的工序尽早结束,即产品尽早完工,于是该问题的数学模型如下

$$T = \min \{ \max \{ E_i \} \} \quad (1)$$

$$\text{s.t. } \min \{ S_i \} \quad (2)$$

$$S_i \geq S_j + T_j \quad (3)$$

$$S_i \geq \max(S_r + T_r) \quad (4)$$

$$S_a = S_b = \dots = S_c \quad E_a = E_b = \dots = E_c \quad (5)$$

式中  $P_i$ ——产品  $p$  上的工序;

$T_i$ —— $P_i$  的连续加工时间;

$S_i$ —— $P_i$  的开始加工时间;

$E_i$ —— $P_i$  的结束加工时间;

$P_j$ —— $P_i$  的同设备紧前工序;

$P_r$ —— $P_i$  的工艺紧前工序。

式(1)表示使产品最后完工工序的结束时间尽可能小;式(2)表示每个工序的开始加工时间尽可能早;式(3)和式(4)表示每个工序的开始加工时间必须大于等于其所有同设备紧前工序和工艺紧前工序的结束时间;式(5)表示本文将相关设备  $a, b, \dots, c$  共同加工的多设备工序  $P_i$  设计成分别在相关设备  $a, b, \dots, c$  上同时处理的多个虚拟工序  $X_a, Y_b, \dots, Z_c$ 。其中,  $1 \leq a, b, \dots, c \leq m$  且  $a \neq b \neq \dots \neq c$ ,  $1 \leq i \leq n$ 。

### 1.2 问题分析

为了简化工序集中存在多设备工序的复杂单产品调度问题,通过对存在该约束条件的复杂单产品工艺树分析,有如下定义。

定义 1 多设备工序:需要多个相关设备协同完成的工序。

定义 2 标准工序:只需要一个设备处理的工序。

定义 3 虚拟工序:把多设备工序设计成在多个相关设备上同时并行加工的多个虚拟工序。

定义 4 相关虚拟工序:属于同一个多设备工序的虚拟工序,一个多设备工序被设计成多个相关虚拟工序。

定义 5 虚拟工序组:由多个相关虚拟工序组成的一组工序,每一个多设备工序都是由多个相关虚拟工序组成的一个虚拟工序组。

定义 6 虚拟加工工艺树:对复杂单产品,根据标准工序和虚拟工序之间的加工工艺约束形成的树状结构,称之为虚拟加工工艺树。

通过以上定义,本文在产品加工工艺树中,将多设备工序设计成在多个相关设备上同时并行加工的多个相关虚拟工序;使产品加工工艺树变为虚拟加工工艺树;将存在多设备工序的调度问题转化为

一般调度问题, 实现对问题的简化。例如图 1 中多设备工序 P2 需要设备 M1 和 M2 同时加工。本文将多设备工序 P2 设计成在相关设备 M1 和 M2 上同时加工的两个虚拟工序 X2 和 Y2, 即多设备工序 P2 变成一个包含相关虚拟工序 {X2, Y2} 的虚拟工序组, 其中 X2 在 M1 上加工、Y2 在 M2 上加工, 虚拟加工工艺树如图 2 所示。

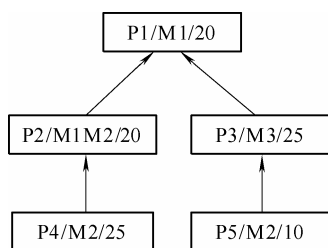


图 1 产品 P 加工工艺树

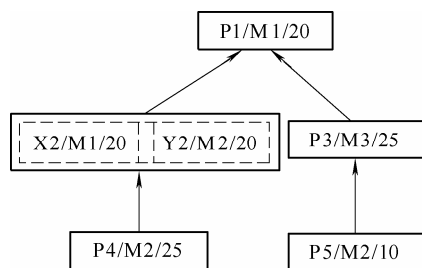


图 2 产品 P 虚拟加工工艺树

为了缩短存在多设备工序的复杂单产品的总加工时间, 降低调度算法的时间复杂度, 本文考虑先确定工序的调度顺序再确定工序在设备上的开始加工时间, 于是提出了确定工序调度顺序的策略和确定工序开始时间的策略。

## 2 确定工序调度顺序的策略

由于本文研究的复杂单产品中存在需要多台设备共同加工的多设备工序, 每个多设备工序被设计成一个包含多个相关虚拟工序的虚拟工序组。当调度虚拟工序组中的相关虚拟工序时, 除了满足工艺紧前和设备紧前约束条件外, 还需要让多个相关虚拟工序在相应的设备上同时开始加工, 所以虚拟工序组中虚拟工序对于加工设备的要求较高。为了控制标准工序提前加工, 过早地占用虚拟工序组的设备时间, 考虑优先调度虚拟工序组中的虚拟工序, 于是, 提出了优先调度虚拟工序的策略。

### 2.1 优先调度虚拟工序的策略

(1) 为了优先调度虚拟工序组中的虚拟工序, 将产品虚拟加工工艺树分解为以虚拟工序组为根结点的子树和剩余标准工序组成的子树, 优先调度以

虚拟工序组为根结点的子树。

(2) 为了让更多的虚拟工序组尽早加工, 当以虚拟工序组为根结点的子树不唯一时, 优先调度包含虚拟工序组个数较多的子树。

(3) 对每个以虚拟工序组为根结点的子树, 优先调度虚拟工序组个数较多的路径上工序。

### 2.2 确定工序调度顺序的具体实现步骤

(1) 从产品虚拟加工工艺树根结点工序开始按层从左到右检索, 如果检索到一个虚拟工序组, 将以该虚拟工序组为根结点的子树从产品虚拟加工工艺树中分解出来, 直到产品虚拟加工工艺树中不存在虚拟工序组为止。

(2) 对步骤(1)得到的以虚拟工序组为根结点的子树分别计算其包含的虚拟工序组个数, 并按包含的虚拟工序组个数由大到小排序各子树。当包含的虚拟工序组个数相同时, 优先排序属于产品虚拟加工工艺树上加工时间较长路径上的子树。

(3) 按步骤(2)得到的排序顺序依次调度以虚拟工序组为根结点的子树。确定以虚拟工序组为根结点的子树上工序的调度顺序的具体方法如下。

1) 确定子树上各路径上虚拟工序组的个数。

2) 按虚拟工序组个数由大到小顺序确定各路径的调度顺序, 选择虚拟工序组个数最多的路径为计划调度路径。

3) 路径不唯一时, 根据长路径策略, 选择子树上加工时间较长的路径为计划调度路径。

4) 找到离计划调度路径上叶结点最近的虚拟工序组, 若该虚拟工序组是叶结点, 则直接调度该虚拟工序组中的虚拟工序; 若不是叶结点, 优先调度计划调度路径上该虚拟工序组的前序工序, 对其他路径上的工序, 根据长路径策略确定其调度顺序。确定完全部前序工序的调度顺序之后, 调度该虚拟工序组中的虚拟工序。

5) 重复步骤 4), 直到计划调度路径上所有工序确定调度顺序。

6) 重复步骤 4)、5), 按路径次序确定该子树上所有工序调度顺序。

7) 判断是否存在其余以虚拟工序组为根结点的子树, 如存在, 转步骤 1); 否则, 转步骤 8)。

8) 结束。

当确定完所有以虚拟工序组为根结点的子树上工序的调度顺序后, 剩余的标准工序按拟关键路径法确定调度顺序。

根据以上优先调度虚拟工序组中虚拟工序的策略实现流程图如图 3 所示。

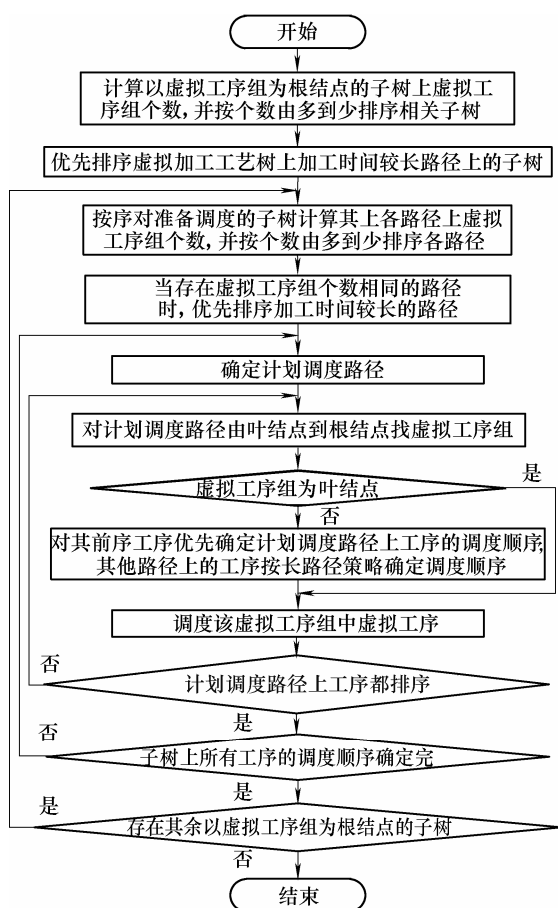


图 3 优先调度虚拟工序的策略流程图

### 3 确定工序开始时间的策略

为了使工序尽早开始加工，采用首次适应调度策略调度标准工序和虚拟工序组中虚拟工序；为了使相关虚拟工序同时开始加工，提出虚拟工序动态调整策略。

#### 3.1 首次适应调度策略

由文献[17]知，首次适应调度策略是工序在满足工艺紧前和设备紧前约束条件下，在相应的设备上由前向后查找第 1 个该工序可加工的空闲时间段进行调度。由于首次适应调度策略不仅实现简单，而且不移动已调度工序，避免破坏已调度虚拟工序组中虚拟工序同时加工的约束条件，因此本文采用首次适应调度策略调度标准工序和虚拟工序组中虚拟工序。

#### 3.2 虚拟工序动态调整策略

采用首次适应调度策略确定虚拟工序组中虚拟工序的开始加工时间时，需要判断相关虚拟工序是否满足同时加工的约束条件，如果不满足，选按照所提出的虚拟工序动态调整策略调整各相关虚拟

工序的开始加工时间。

虚拟工序动态调整策略的思路如下：首先对虚拟工序组中的各虚拟工序采用首次适应调度策略尽早加工；其次当虚拟工序组中各虚拟工序不能同时开始加工时，将相关虚拟工序统一调整到最晚开始加工时间；如果一次后移调整不能使相关虚拟工序同时开始加工，重复以上调整步骤，直到相关虚拟工序同时开始加工。

假设某一多设备工序，被设计成在相关设备  $M_1, M_2, \dots, M_f$  上加工的  $f$  个相关虚拟工序，加工时间为  $T$ 。虚拟工序动态调整策略具体实现步骤如下。

(1) 首先采用首次适应调度策略确定虚拟工序组中各虚拟工序在其设备上的开始加工时间，假设此时各相关虚拟工序的开始加工时间分别为  $S_1, S_2, \dots, S_f$ 。

(2) 判断虚拟工序组中的相关虚拟工序的开始加工时间是否相同，如果所有相关虚拟工序的开始加工时间均相同，转步骤(7)；否则转步骤(3)。

(3) 设  $t = \max\{S_1, S_2, \dots, S_f\}$ ，即  $t$  为虚拟工序组中相关虚拟工序最晚开始加工时间。

(4) 判断其他相关虚拟工序的加工设备上  $\{t, t+T\}$  时间段是否空闲：假设设备  $M_i$  ( $1 \leq i \leq f$ ) 上已经调度工序的开始加工时间和结束加工时间分别为  $S_{i1}, S_{i2}, \dots, S_{iu}$  和  $E_{i1}, E_{i2}, \dots, E_{iu}$  (不包括当前虚拟工序组中相关虚拟工序)，判断设备  $M_i$  在时间段  $\{t, t+T\}$  是否空闲，只需要判断已调度工序的开始加工时间  $S_{ij}$  和结束加工时间  $E_{ij}$  是否在  $\{t, t+T\}$  时间段内即可，其中 ( $1 \leq j \leq u$ )，共有以下 3 种情况：①  $S_{ij} < t, E_{ij} \leq t$ ；②  $t \leq S_{ij} < t+T$  或  $t \leq E_{ij} < t+T$ ；③  $S_{ij} \geq t+T, E_{ij} \geq t+T$ 。

当情况①或情况③时，设备  $M_i$  上  $\{t, t+T\}$  时间段空闲。

当情况②时，设备  $M_i$  上  $\{t, t+T\}$  时间段不空闲。

(5) 若相关虚拟工序的加工设备均空闲，后移相关虚拟工序使它们在  $t$  时刻同时开始；转步骤(7)；否则转步骤(6)。

(6) 将步骤(4)中  $\{t, t+T\}$  非空闲设备上加工的相关虚拟工序从  $t$  时刻向后寻找尽早开始加工的时刻  $t_1$ ，令  $t = t_1$ ，转步骤(4)。

(7) 虚拟工序动态调整结束。

根据以上虚拟工序动态调整策略实现步骤，相应的策略流程图如图 4 所示。

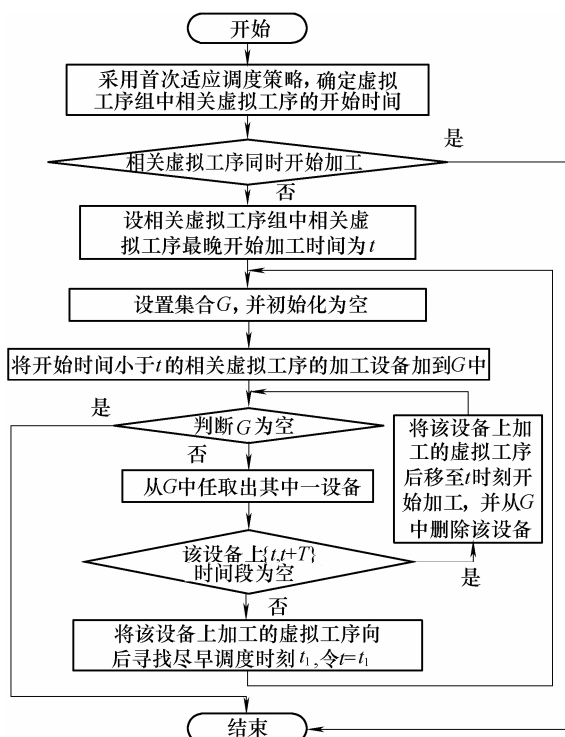


图4 虚拟工序动态调整策略流程图

## 4 算法设计

根据以上分析, 对于存在多设备共同加工单个工序的综合调度算法设计如下。

- (1) 根据产品加工信息构造加工工艺树。
- (2) 将多设备工序设计成虚拟工序组, 使加工工艺树变为虚拟加工工艺树。
- (3) 将虚拟加工工艺树分解为以虚拟工序组为根结点的子树和剩余标准工序组成的子树。
- (4) 采用优先调度虚拟工序组中虚拟工序的策略, 确定以虚拟工序组为根结点的子树的调度顺序和各子树上工序的调度顺序。
- (5) 采用拟关键路径法确定剩余标准工序的调度顺序。
- (6) 按所确定的调度顺序采用首次适应调度策略确定工序的开始加工时间, 若是标准工序, 确定的开始加工时间不变; 若是虚拟工序组, 需要判断相关虚拟工序是否满足同时加工的约束条件, 若不满足, 采用虚拟工序动态调整策略调整各相关虚拟工序的开始加工时间, 使其同时开始加工。
- (7) 判断是否存在其余未调度的工序, 若存在转步骤(6); 若不存在转步骤(8)。
- (8) 输出甘特图。
- (9) 结束。

算法流程图如图 5 所示。

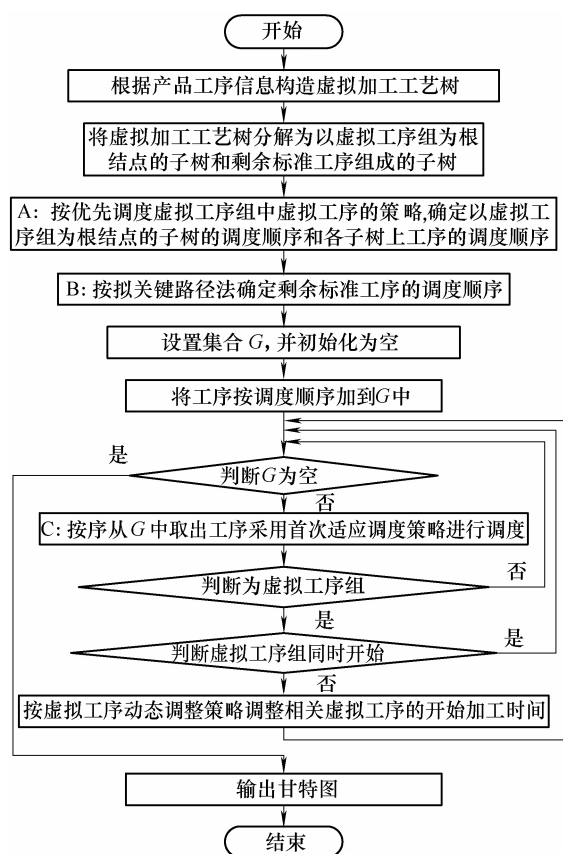


图5 算法流程图

## 5 算法复杂度分析

设产品工序数是  $n$ , 其中多设备工序数是  $g$ ,  $g$  个多设备工序对应的虚拟工序数为  $w$ , 设备数是  $m$ , 算法主要有如下操作。

(1) 确定以虚拟工序组为根结点的子树的调度顺序。设以虚拟工序组为根结点的子树的个数为  $r$  ( $1 \leq r \leq g \leq n$ ), 根据虚拟工序组个数对  $r$  个以虚拟工序组为根结点的子树进行排序, 使用冒泡排序法排序的操作次数为  $r^2$ 。当子树包含的虚拟工序组个数相同时, 优先排序属于虚拟加工工艺树上加工时间较长路径上的子树。最坏的情况下  $r$  个子树包含的虚拟工序组数均相等, 此时需要比较  $r$  条路径的加工时间, 由文献[17]计算各路径长度的复杂度是  $O(n^2)$ , 采用冒泡排序法排序的操作次数为  $r^2$ 。所以对  $r$  个以虚拟工序组为根结点的子树进行排序的时间复杂度为  $O(n^2)$ 。

(2) 确定以虚拟工序组为根结点的子树上工序的调度顺序。有  $n/r$  个工序的子树上最多有  $n/r-1$  条路径, 使用冒泡排序法比较各路径上虚拟工序组个数的操作次数为  $(n/r-1)^2$ 。当不同路径上的虚拟工序组个数相同时, 按长路径策略优先选择加工时间

较长路径上的工序。最坏的情况下  $n/r-1$  条路径上虚拟工序组数均相等, 此时需要比较  $n/r-1$  条路径的加工时间, 由文献[17]计算各路径长度的复杂度是  $O((n/r-1)^2)$ , 采用冒泡排序法比较路径加工时间的操作次数为  $(n/r-1)^2$ 。所以确定一个以虚拟工序组为根结点的子树上工序调度顺序的复杂度为  $O((n/r-1)^2)$ , 确定  $r$  个以虚拟工序组为根结点的子树上工序的调度顺序的算法复杂度为  $O(r(n/r-1)^2)$ , 由于  $1 \leq r \leq n$ , 所以确定所有以虚拟工序组为根结点的子树上工序的调度顺序的算法复杂度为  $O(n^2)$ 。

(3) 确定剩余标准工序的调度顺序。由于产品总工序数是  $n$ , 其中多设备工序数为  $g$ , 确定完所有以虚拟工序组为根结点的子树上工序后, 剩余的标准工序数最多为  $n-g$ , 因此, 采用拟关键路径法确定  $n-g$  个标准工序的调度顺序的算法复杂度为  $O((n-g)^2)$ , 由于  $1 \leq g \leq n$ , 所以确定  $n-g$  个标准工序的调度顺序的算法复杂度为  $O(n^2)$ 。

(4) 确定标准工序的开始加工时间。采用首次适应调度策略确定标准工序的开始加工时间, 在每个设备上调度时, 需要比较的次数平均为已经调度工序数/设备数。对于  $n-g$  个标准工序, 最坏情况为全部虚拟工序的调度顺序均确定完毕, 然后确定全部标准工序的调度顺序, 由于虚拟工序数为  $w$ , 设备数为  $m$ , 所以在每个设备上需要比较的次数最多为  $w/m, w/m+1, w/m+2, \dots, w/m+(n-g)/m-1$  次, 即每个设备上需要比较的总次数为  $\{[w/m+w/m+(n-g)/m-1] \times (n-g)/m\}/2 = [(n-g)^2 + (2w-m)n - g(2w-m)]/2m^2$ , 在  $m$  个设备上需要比较的次数最多为  $[(n-g)^2 + (2w-m)n - g(2w-m)]/2m$ , 由于  $1 < m \ll n$ , 所以确定所有标准工序开始加工时间的复杂度为  $O(n^2)$ 。

(5) 确定虚拟工序组中虚拟工序的开始加工时间。采用首次适应调度策略确定虚拟工序组中虚拟工序的开始加工时间, 最坏情况为全部标准工序及  $g$  个虚拟工序组的前  $g-1$  个均已调度完毕, 且  $g$  个虚拟工序组中虚拟工序数均为  $m$ , 则已调度的虚拟工序数为  $m(g-1)$ , 已调度的标准工序数为  $n-g$ , 所以已调度工序的总数为  $n-g+m(g-1)$ , 此时平均每个加工设备上已调度工序的总数为  $[n-g+m(g-1)]/m$ 。于是每个相关虚拟工序最多需要比较的次数为  $[n-g+m(g-1)]/m$ ,  $m$  个相关虚拟工序需要比较的次数最多为  $n-g+m(g-1)$ 。所以, 采用首次适应调度策略确定一个虚拟工序组开始加工时间的操作次数最

多为  $n-g+m(g-1)$ 。

判断相关虚拟工序是否满足同时开始加工的约束条件,  $m$  个相关虚拟工序的开始加工时间的比较次数为  $C_m^2$ , 当不满足时, 采用虚拟工序动态调整策略后移调整虚拟工序的开始加工时间。

设第  $g$  个虚拟工序组的加工时间为  $T$ , 相关虚拟工序最晚开始加工时间为  $t$ , 判断其余相关虚拟工序的加工设备上  $(t, t+T)$  时间段是否空闲: 由于每个设备上已调度工序数最多为  $[n-g+m(g-1)]/m$ , 所以每个相关设备需要比较次数最多为  $[n-g+m(g-1)]/m$ ; 又由于每次比较需要判断工序开始和结束时间是否在  $(t, t+T)$  时间段内, 最多需要操作 4 次, 于是每个相关设备需要比较的操作次数最多为  $4[n-g+m(g-1)]/m$ ;  $m-1$  个相关设备需要比较操作的次数最多为  $4(m-1)[n-g+m(g-1)]/m$ 。由于相关虚拟工序最晚开始加工时间  $t$  一定在  $m$  个加工设备上的空闲时间段或设备末尾, 所以虚拟工序后移寻找同时加工时间的次数最多为  $n-g+m(g-1)$ 。因此, 采用虚拟工序动态调整策略后移调整虚拟工序开始加工时间的操作次数最多为  $4\{(m-1)[n-g+m(g-1)]/m\} \times [n-g+m(g-1)] = 4(m-1)[n-g+m(g-1)]^2/m$ 。

综上所述, 确定一个虚拟工序组开始加工时间的操作总次数最多为  $n-g+m(g-1) + C_m^2 + 4(m-1) \times [n-g+m(g-1)]^2/m$ 。由于虚拟工序组总数为  $g$ , 所以确定全部虚拟工序组中虚拟工序的开始加工时间, 需要比较的总次数最多为  $g\{n-g+m(g-1) + C_m^2 + 4 \times (m-1)[n-g+m(g-1)]^2/m\} < g\{4[n-g+m(g-1)]^2 + n-g+m(g-1) + C_m^2\}$ 。由于  $1 < m \ll n$ , 所以确定全部虚拟工序开始加工时间的复杂度为  $O(gn^2)$ 。

综合以上五点, 本文提出算法的时间复杂度为  $O(gn^2)$ , 因为  $0 \leq g \leq n$ , 本文提出算法的最大时间复杂度为  $O(n^3)$ , 当  $g$  值较小时, 该算法的时间复杂度一般为  $O(n^2)$ 。

## 6 实例分析

为了方便读者理解本文算法, 下面通过实例分析说明。设有产品 A 其加工工艺树如图 6 所示, 图中长方框的内容为工序名/设备名/加工时间。例如工序 A4 在设备 M4 上加工, 加工时间为 20 单位时间。其中有多设备工序 {A3, A8, A13, A14, A17, A20, A23, A25}。构造相应的虚拟加工工艺树如图 7 所示。

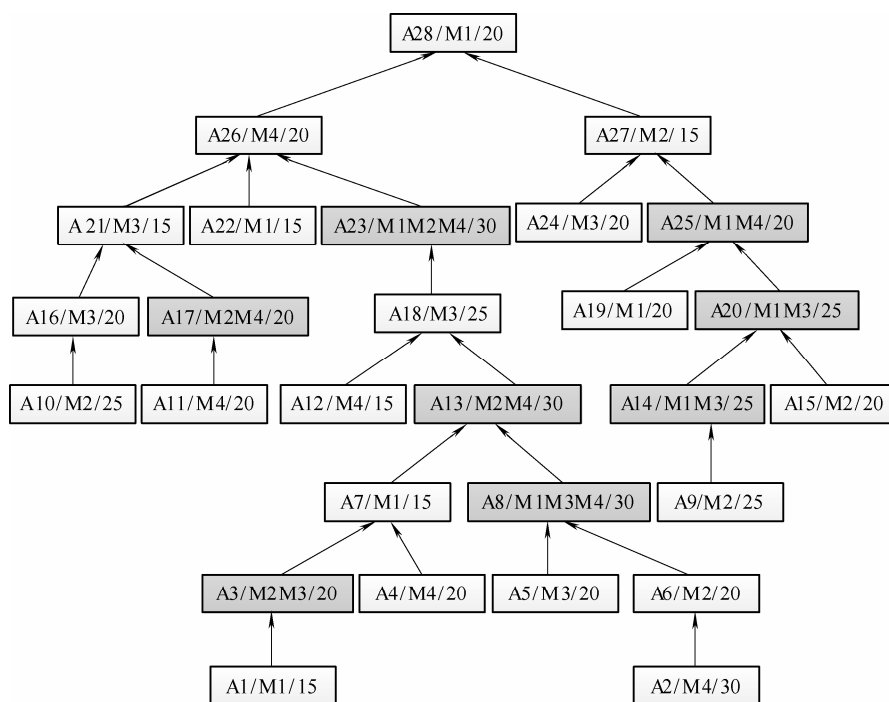


图 6 产品 A 加工工艺树

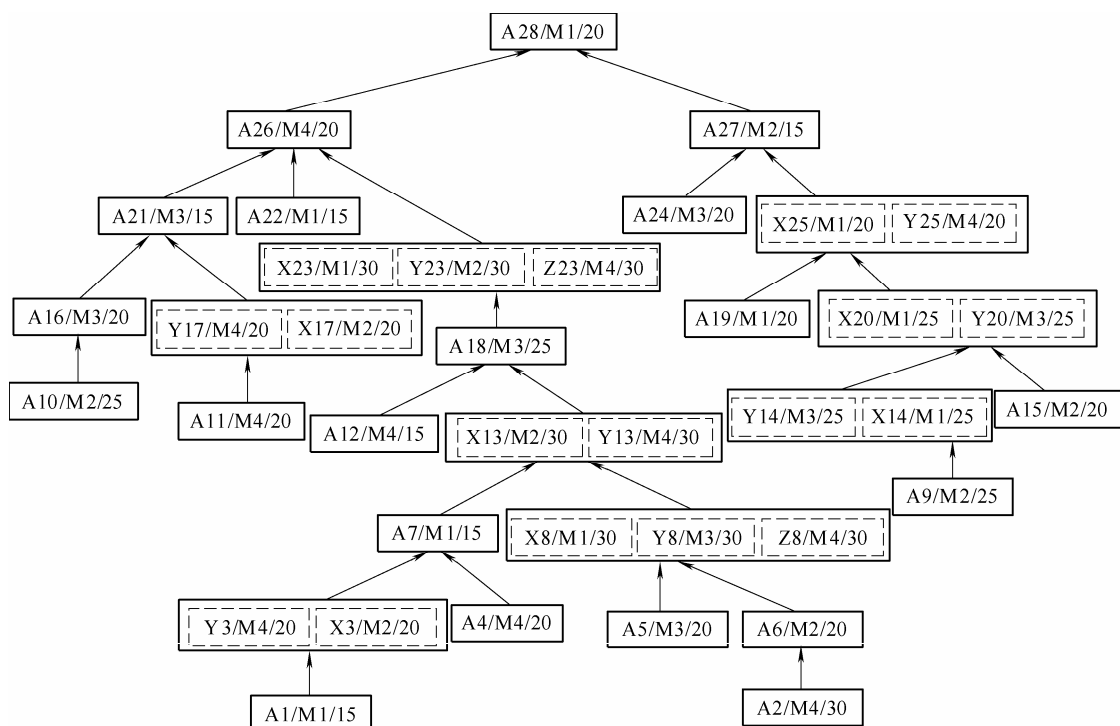


图 7 产品 A 虚拟加工工艺树

图 6 中多设备工序 A3,A8,A13,A14,A17,A20,A23 和 A25 分别对应图 7 虚拟工序组 {X3,Y3}, {X8,Y8,Z8}, {X13,Y13}, {X14,Y14}, {X17,Y17}, {X20,Y20}, {X23,Y23,Z23} 和 {X25,Y25}。

从虚拟加工工艺树根节点工序 A28 开始按层从左到右检索，将检索到的虚拟工序组为根节点的子树从产品虚拟加工工艺树中分解出来。首先检索到虚拟工序组 {X23,Y23,Z23}，将以该虚拟工序组为

根节点的子树 {A1,A2,{X3,Y3}, A4, A5, A6, A7, {X8,Y8,Z8}, A12, {X13,Y13}, A18,{X23,Y23,Z23}} 从虚拟加工工艺树中分解出来，图 8 为子树 1。同理继续在产品 A 的虚拟加工工艺树上检索虚拟工序组，分别得到以虚拟工序组 {X25,Y25} 和 {X17,Y17} 为根节点的子树 2 和子树 3，如图 9 和图 10 所示。剩余的标准工序 {A10, A16, A21, A22, A24, A26, A27, A28} 组成的子树 4，如图 11 所示。下面优先调度以

虚拟工序组为根结点的子树，再调度剩余标准工序组成的子树。

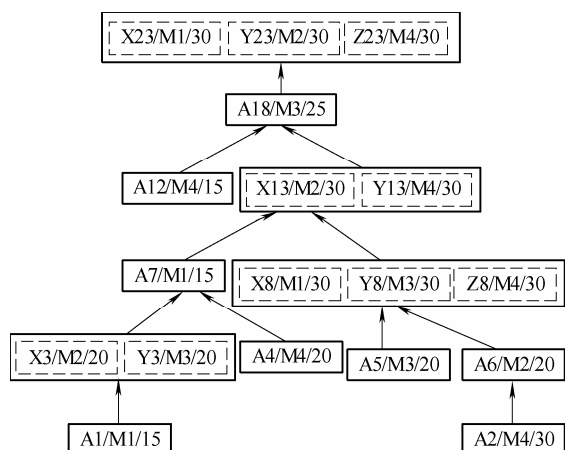


图 8 子树 1

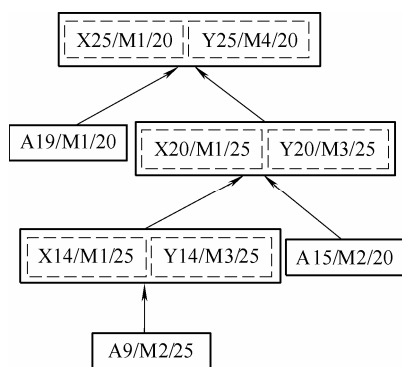


图 9 子树 2

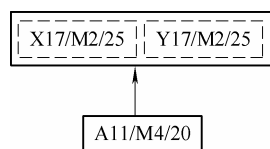


图 10 子树 3

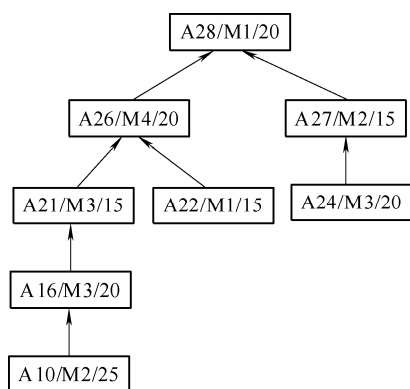


图 11 子树 4

根据包含的虚拟工序组个数多少确定以虚拟工序组为根结点的子树的调度顺序，由于三个子树上虚拟工序组个数分别为 4、3、1，所以调度次序为子树 1、子树 2 和子树 3。

对于子树 1，路径 L1: A2,A6,{X8,Y8,Z8},{X13,

Y13},A18,{X23,Y23,Z23}和路径 L2: A1,{X3,Y3},A7,{X13,Y13},A18,{X23,Y23,Z23}上的虚拟工序组个数都是 3，由于 L1 的加工时间和 165 大于 L2 的时间和 135，所以 L1 为优先调度的计划调度路径。于是按本文算法子树 1 上工序的调度顺序如下: A2,A6,A5,{X8,Y8,Z8},A1,{X3,Y3},A4,A7,{X13,Y13},A12,A18,{X23,Y23,Z23}。

同理，根据图 9，子树 2 上工序的调度顺序为 A9,{X14,Y14},A15,{X20,Y20},A19,{X25,Y25}。根据图 10，子树 3 上工序的调度顺序为 A11,{X17,Y17}。

对图 11 所示的剩余标准工序组成的子树，采用文献[12]中拟关键路径法确定的工序调度顺序如下: A10,A16,A21,A22,A26,A24,A27,A28。

所以产品 A 的调度顺序如下: A2,A6,A5,{X8,Y8,Z8},A1,A4,{X3,Y3},A7,{X13,Y13},A12,A18,{X23,Y23,Z23},A9,{X14,Y14},A15,{X20,Y20},A19,{X25,Y25},A11,{X17,Y17},A10,A16,A21,A22,A26,A24,A27,A28。

按首次适应调度策略确定各工序的开始加工时间及虚拟工序组中虚拟工序调整过程如表 1 所示。相应的产品 A 调度的甘特图如图 12 所示，总加工时间为 260，其中虚拟工序用虚框表示。

如果不分解产品虚拟加工工艺树，对于产品虚拟加工工艺树，优先调度包含虚拟工序组个数多的路径上的工序，若包含的虚拟工序组个数相同时，采用长路径策略优先调度加工时间长路径上的工序，确定完所有工序的调度顺序后采用首次适应调度策略确定标准工序和虚拟工序的开始加工时间，当虚拟工序组中虚拟工序不满足同时开始加工的约束条件时，采用虚拟工序动态调整策略进行调整，产品加工的甘特图如图 13 所示。

由于本文解决的是综合调度中的新问题，目前只能对比不采用本文算法的情况。由图 12 和图 13 可看出本文所提出的算法更优。主要原因如下: 本文算法先把产品虚拟加工工艺树分解为以虚拟工序组为根结点的子树和剩余标准工序组成的子树，优先调度以虚拟工序组为根结点的子树，最大限度地控制了标准工序提前加工，过早地占用虚拟工序组的设备时间，进而缩短产品总加工时间。例如，采用本文算法在本例子中先调度以虚拟工序组 {X25,Y25} 为根结点的子树 2，再调度标准工序 A10,A16,A21,A22,A26，使得这些标准工序不能干扰子树 2 上的虚拟工序组。对比图 12 和图 13，多设备工序 A25 的开始加工时间由 145 延后至 245，导致产品总加工时间由 260 延后至 300，主要原因是 A26 抢占了多设备工序 A25 的加工设备。



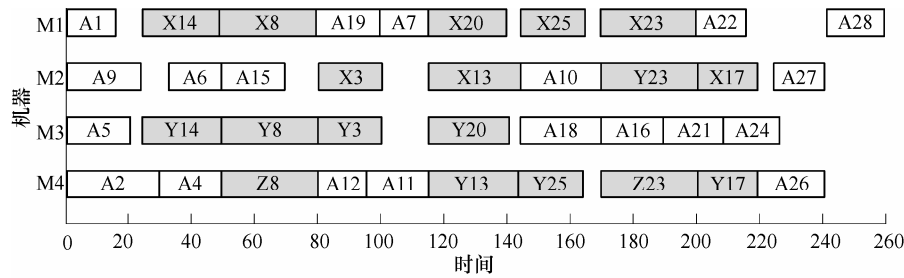


图 12 产品 A 甘特图(总用时 260)

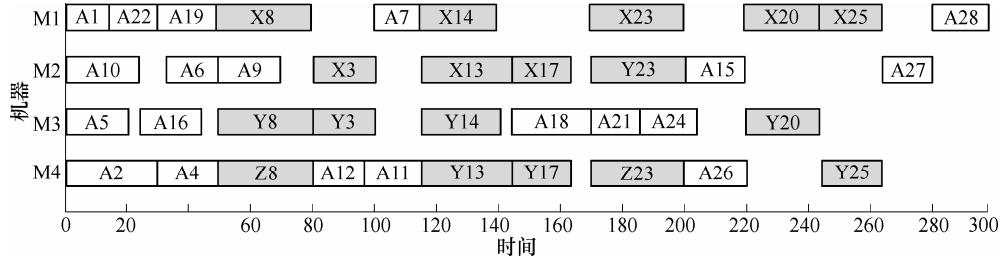


图 13 产品 A 甘特图(总用时 300)

表 1 产品 A 中工序确定开始时间表

工序名	确定工序开始时间	虚拟工序调整后时间	备注
A2	0	—	—
A6	30	—	—
A5	0	—	—
X8	50		
Y8	50	50	A8 无调整
Z8	50		
A1	0	—	—
A4	30	—	—
X3	50	80	A3 调整 2 次
Y3	20		
A7	100	—	—
X13	115	115	A13 无调整
Y13	115		
A12	80	—	—
A18	145	—	—
X23	170		
Y23	170	170	A23 无调整
Z23	170		
A9	0	—	—
X14	25	25	A14 无调整
Y14	25		
A15	50	—	—
X20	115	115	A20 调整 1 次
Y20	100		
A19	80	—	—
X25	140	145	A25 调整 1 次
Y25	145		
A11	95	—	—
X17	200	200	A17 无调整
Y17	200		
A10	145	—	—
A16	170	—	—
A21	190	—	—
A22	200	—	—
A26	220	—	—
A24	205	—	—
A27	225	—	—
A28	240	—	—

## 7 结论

本文针对工序集中存在多设备工序的综合调度问题，提出了把单个多设备工序设计成在多个相关设备上同时并行加工的多个虚拟工序的综合调度算法。

(1) 由于该算法优先调度虚拟工序组，避免了标准工序过早地占用加工设备，使存在多设备工序的产品尽早完工。

(2) 由于提出的虚拟工序动态调整策略，可较方便地确定虚拟工序组尽早开始时间，控制了此复杂问题的算法复杂度。

因此本文提出的调度算法不仅能较好地解决工序集中存在多设备工序的综合调度问题，而且算法简单实用，可以为进一步研究复杂综合调度问题提供参考。

## 参 考 文 献

- [1] ZHANG H P, GEN M. Multistage-based genetic algorithm for flexible job-shop scheduling problem[J]. Complexity International, 2005, 11: 223-232.
- [2] YOO Y B, HELZER D. The parallel genetic algorithm for the job shop scheduling problem[C]//Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing, 2003, 7: 249-254.
- [3] ENGIN O, CERAN G, YILMAZ M K. An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems[J]. Applied Soft Computing Journal, 2011, 11(3): 3056-3065.
- [4] MIRSANEI H S, ZANDIEH M, MOAYED M J, et al. A simulated annealing algorithm approach to hybrid flow

- shop scheduling with sequence-dependent setup times [J]. Journal of Intelligent Manufacturing, 2011, 22(6): 965-978.
- [5] GLOVER F. Future paths for integer programming and links to artificial intelligence[J]. Computers & Operations Research, 1986, 13(5): 533-549.
- [6] FOO Y S, TAKEFUJI Y. Stochastic neural networks for solving job-shop scheduling[C/CD]//Problem Representation, Stochastic Neural Networks for Solving Job-shop Scheduling: Part2. Architecture and Simulations. IEEE International Conference on Neural Networks, San Diego, 1998.
- [7] 熊禾根, 李建军, 孔建益. 考虑工序相关性的动态 Job Shop 调度问题启发式算法[J]. 机械工程学报, 2006, 42(8): 50-55.
- XIONG Hegen, LI Jianjun, KONG Jianyi. Heuristic method for dynamic job shop scheduling problem with operation relativity[J]. Chinese Journal of Mechanical Engineering, 2006, 42(8): 50-55.
- [8] 闫利军, 李宗斌, 卫军胡, 等. 一种新的混合优化算法及其在车间调度中的应用[J]. 自动化学报, 2008, 34(5): 604-608.
- YAN Lijun, LI Zongbin, WEI Junhu, et al. A new hybrid optimization algorithm and its application in job shop scheduling[J]. Acta Automatica Sinica, 2008, 34(5): 604-608.
- [9] 张长胜, 孙吉贵, 杨轻云, 等. 一种求解车间调度的混合算法[J]. 自动化学报, 2009, 35(3): 332-336.
- ZHANG Changsheng, SUN Jigui, YANG Qingyun, et al. A hybrid algorithm for flow shop scheduling problem[J]. Acta Automatica Sinica, 2009, 35(3): 332-336.
- [10] LIN T L, HORONG S J, KAO T W, et al. An efficient job-shop scheduling algorithm based on particle swarm optimization[J]. Expert Systems with Applications, 2010, 37(3): 2629-2636.
- [11] MEERAN S, MORSHED M S. A hybrid genetic tabu search algorithm for solving job shop scheduling problems: A case study[J]. Journal of Intelligent Manufacturing, 2012, 23(4): 1063-1078.
- [12] 谢志强, 刘胜辉, 乔佩利. 基于 ACPM 和 BFSM 的动态 Job-Shop 调度算法[J]. 计算研究与发展, 2003, 40(7): 977-983.
- XIE Zhiqiang, LIU Shenghui, QIAO Peili. Dynamic job-shop scheduling algorithm based on ACPM and BFSM[J]. Journal of Computer Research and Development, 2003, 40(7): 977-983.
- [13] 谢志强, 辛宇, 杨静. 基于设备空闲事件驱动的综合调度算法[J]. 机械工程学报, 2011, 47(11): 139-147.
- XIE Zhiqiang, XIN Yu, YANG Jing. Integrated scheduling algorithm based on event-driven by machines' idle [J]. Journal of Mechanical Engineering, 2011, 47(11): 139-147.
- [14] 谢志强, 张伟涛, 杨静. 前移存在调整时间综合调度工序的算法[J]. 机械工程学报, 2012, 48(12): 169-177.
- XIE Zhiqiang, ZHANG Weitao, YANG Jing. Algorithm of moving integrated scheduling procedures with set-up time forward [J]. Journal of Mechanical Engineering, 2012, 48(12): 169-177.
- [15] 谢志强, 邵侠, 杨静. 存在设备无关延迟约束的综合柔性调度算法[J]. 机械工程学报, 2011, 47(4): 177-185.
- XIE Zhiqiang, SHAO Xia, YANG Jing. Algorithm for integrated flexible scheduling with device- independence deferred constraint[J]. Journal of Mechanical Engineering, 2011, 47(4): 177-185.
- [16] 谢志强, 李志敏, 郝淑珍, 等. 工序间存在零等待约束的复杂产品调度研究[J]. 自动化学报, 2009, 35(7): 983-989.
- XIE Zhiqiang, LI Zhimin, HAO Shuzhen, et al. Study on complex product scheduling problem with no-wait constraint between operations[J]. Acta Automatica Sinica, 2009, 35(7): 983-989.
- [17] 谢志强, 杨静, 周勇, 等. 基于工序集动态关键路径多产品制造调度算法[J]. 计算机学报, 2011, 34(2): 406-412.
- XIE Zhiqiang, YANG Jing, ZHOU Yong, et al. Dynamic critical paths multi-product manufacturing scheduling algorithm based on operation set[J]. Chinese Journal of Computers, 2011, 34(2): 406-412.

作者简介: 谢志强, 男, 1962 年出生, 博士后, 教授, 博士研究生导师, CMES 高级会员。主要研究方向为企业智能计算与调度优化。

E-mail: xzq011@tom.com

齐永红, 女, 1988 年出生。主要研究方向为企业智能计算与调度优化。

E-mail: vivian12250@163.com

杨静, 女, 1962 年出生, 博士, 教授, 博士研究生导师。主要研究方向为 CIMS、数据库与知识工程。

E-mail: yangjing@hrbeu.edu.cn