

## 考虑后续工序的择时综合调度算法

谢志强<sup>1</sup> 张晓欢<sup>1</sup> 辛宇<sup>1</sup> 杨静<sup>2</sup>

**摘要** 针对目前综合调度算法不能兼顾产品工艺树中并行工序的并行性和串行工序之间紧密度,影响调度结果的问题,提出考虑后续工序的择时综合调度算法.该算法提出工序序列排序策略,从工艺树的整体结构出发,将其划分成若干内部工序只具有串行关系的工序序列,并按路径长度从长到短的顺序确定其调度次序;提出择时调度策略和考虑后续工序策略,根据工艺树自身特点,来自不同工序序列的并行工序的不同组合方案中,选择最接近调度目标的方案作为工序调度方案,若该工序调度方案不唯一,则在其中选择该工序加工开始时间最早的调度方案.该算法既保证了工序的并行处理,又提高了串行工序的紧密度,优化了综合调度的结果.最后通过实例说明本文算法对解决综合调试问题具有普遍意义.

**关键词** 工序序列, 后续工序, 择时, 综合调度

**引用格式** 谢志强, 张晓欢, 辛宇, 杨静. 考虑后续工序的择时综合调度算法. 自动化学报, 2018, 44(2): 344–362

**DOI** 10.16383/j.aas.2018.c160562

## Time-selective Integrated Scheduling Algorithm Considering Posterior Processes

XIE Zhi-Qiang<sup>1</sup> ZHANG Xiao-Huan<sup>1</sup> XIN Yu<sup>1</sup> YANG Jing<sup>2</sup>

**Abstract** Integrated scheduling algorithms currently neglected the compactness of serial processes when handling a general integrated scheduling problem, and it influenced the scheduling result. Aiming at this problem, an time-selective integrated scheduling algorithm considering posterior processes was presented. The strategy of process sequence sorting was proposed. From the overall structure of the process tree, it was divided into several sequence of processes in which the processes only had a serial relationship. According to the path length to determined the order of its scheduling. The strategy of time-selective and considering posterior processes was proposed. According to the characteristics of the process tree, selected the most close to the scheduling objectives as a process scheduling scheme from the different combination of parallel process from different process sequence. If the process scheduling scheme was not unique. selected the process scheduling scheme in which The processing start time of the process was the earliest. This algorithm promises to proceed together the parallel processing of processes, and effectively raises the compactness of serial processes. The results of integrated scheduling are optimized. Finally illustrated by examples.

**Key words** Process sequence sorting, posterior processes, time-selective, integrated scheduling

**Citation** Xie Zhi-Qiang, Zhang Xiao-Huan, Xin Yu, Yang Jing. Time-selective integrated scheduling algorithm considering posterior processes. *Acta Automatica Sinica*, 2018, 44(2): 344–362

收稿日期 2016-08-19 录用日期 2017-05-04  
Manuscript received August 19, 2016; accepted May 4, 2017  
国家自然科学基金 (61370086, 61370083, 61602133, 61672179, 6172160), 黑龙江省教育厅科技项目 (12531105), 黑龙江省博士后科研启动项目 (LBH-Q13092), 中国博士后资助项目 (2016M591541), 黑龙江省博士后资助项目 (LBH-Z15096), 高等学校博士学科点专项 (博导类) 科研基金 (20122304110012) 资助  
Supported by National Natural Science Foundation of China (61370086, 61370083, 61602133, 61672179, 6172160), the Science and Technology Project of Heilongjiang Provincial Department of Education (12531105), the Heilongjiang Scientific Research foundation for the Postdoctoral (LBH-Q13092), the China Postdoctoral Science Foundation (2016M591541), the Heilongjiang Scientific Research Program for the Postdoctoral (LBH-Z15096), and Research Fund for the Doctoral Program of Higher Education (20122304110012)  
本文责任编辑 王红卫  
Recommended by Associate Editor WANG Hong-Wei  
1. 哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080 2. 哈尔滨工程大学计算机科学与技术学院 哈尔滨 150080  
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 150080

传统车间调度一般分为 Flow-shop (大批量相同产品采取的流水调度) 调度和 Job-shop (多品种小批量产品采取的车间调度) 调度, 两类调度中产品均采用先加工后装配的制造方式. 因此, 研究方向主要分为单一加工调度<sup>[1-5]</sup> 和单一装配调度<sup>[6-7]</sup>. 上述两种调度算法在调度时, 均将产品拆分成若干工件, 集中生产工件, 生产完毕集中装配. 这种先加工后装配的生产方式在大规模生产时代比较常见. 原因是订单中产品数量较大, 集中生产会产生大量库存, 使得生产活动与装配活动之间存在大量并行时间, 从而缩短产品生产周期. 然而当今社会对产品的需求不断向个性化和多样化转变, 多品种小批量, 甚至单件产品的定单越来越多, 产品库存极少甚至无库存, 先加工后装配必然割裂产品加工和装配内在的并行关系. 例如, 一个产品可拆分成若干工件, 加工过程中某些先加工完毕的工件可直接装配在一起, 且该装配过程可与当前未加工完毕工件的加工活动

同时进行, 而不需要等待所有工件加工完毕再进行装配, 提高了加工与装配的并行性, 缩短产品生产周期. 这样, 同时考虑产品的加工与装配的调度称之为综合调度<sup>[8-18]</sup>. 其更适合在单件小批订单中应用. 目前, 学者们已在综合调度领域做了大量的研究, 例如, 具有约束关系的复杂产品柔性综合调度算法<sup>[13]</sup>, 工序间具有紧密衔接关系的多产品综合调度算法<sup>[14]</sup>, 基于设备驱动和动态短路径的综合调度算法<sup>[15]</sup>, 基于相同设备的动态多产品综合调度算法<sup>[16]</sup>等等. 其中在研究单车间一般综合调度问题时, 算法主要从工序和设备两个角度进行了研究.

在基于工序的研究中, 目前调度效果最好的方法是文献 [10] 提出的动态关键路径算法, 调度工序时, 在纵横兼顾的基础上, 进一步优化了纵向调度; 在基于设备的研究<sup>[11-12]</sup> 中, 目前调度效果最好的方法是文献 [12] 提出带有回退抢占策略的设备驱动调度算法, 其最大化了“设备忙”原则, 减少设备空闲时间, 且具有回退抢占策略强调了长路径在调度中的重要影响. 但这两方面的研究所提算法均以从产品加工工艺树的一端向另一端逐步推进的方式进行调度, 调度工序时以当前已调度工序所组成的部分产品调度计划为基础, 以工序尽早加工为原则. 算法缺点为: 1) 没有充分考虑已调度的并行工序抢占加工设备等因素的制约, 导致出现串行工序之间紧密度差, 影响调度结果的问题; 2) 没有考虑先加工工序的调度对后加工工序的影响, 导致出现工序之间并行性差的问题.

针对以上两个问题, 提出考虑后续工序的择时综合调度算法. 1) 根据工序序列排序策略确定工艺树中工序的调度次序; 2) 调度工序时, 在已调度工序所形成的部分产品调度计划中为调度工序确定若干“准调度时间点”; 3) 在每个“准调度时间点”上对该工序进行试调度, 其中采用择时调度策略对该调度工序影响当前已调度工序情况进行分析, 采用考虑后续工序策略对该调度工序影响未调度工序情况进行分析; 4) 综合考虑上述两种情况, 确定该调度工序的加工开始时间点. 该算法充分结合产品加工工艺树特点, 调度工序时与已调度工序和其后续工序相配合, 有效地提高了串行工序的紧密度和并行工序的并行性.

## 1 问题模型描述

综合调度问题研究的生产模式为产品边加工边装配, 综合调度算法是寻找使产品加工总用时最少的调度方法, 本文所解决问题为单产品综合调度问题, 文中工序是加工工序与装配工序的统称, 设备是加工设备和装配设备的统称, 调度以工序为基本单位.

综合调度问题一般满足以下要求: 1) 每道工序指定唯一加工设备; 2) 不存在相同设备; 3) 每台设备任一时刻均最多有一道工序加工; 4) 工序的加工过程不能被间断.

设产品由  $n$  道工序组成, 在  $m$  台设备上加工, 若工序表示为  $i$ , 则其所解决综合调度问题涉及参数如表 1 所示, 本文根据描述需要分别在后文中将工序设为  $A_i (i \in \mathbf{N}), B_i (i \in \mathbf{N}), C_i (i \in \mathbf{N}), W_i (i \in \mathbf{N}), P_i (i \in \mathbf{N}), \beta_i (i \in \mathbf{N}), \alpha_i (i \in \mathbf{N}), A$  和  $B$  等, 上述工序符号与表 1 中符号  $i$  同义, 在文中有相应说明.

表 1 综合调度问题参数表

Table 1 Parameter list of integrated scheduling problem

参数	含义
$M_i$	工序 $i$ 的加工设备
$T_{Si}$	工序 $i$ 的加工开始时间
$T_i$	工序 $i$ 的加工用时
$T_{Ei}$	工序 $i$ 的加工结束时间
$T_{Mki}$	调度工序 $i$ 后第 $k$ 台设备上当前加工完成时间, $1 \leq k \leq m$
$T_{Mi}$	调度工序 $i$ 后当前最晚加工完成的已调度工序加工完成时间
$T_{Tij}$	工序 $i$ 的第 $j$ 个“准调度时间点”, $j \geq 1$
$T_{Ti}$	工序 $i$ 的“准调度时间点”集合
$F_{ij}$	工序 $i$ 在工艺树中的第 $j$ 个紧前工序, $j \geq 1$
$N_{if}$	工序 $i$ 在工艺树中的第 $f$ 个紧后工序, $f \geq 1$
$F_{Mi}$	工序 $i$ 在其加工设备上的紧前工序
$N_{Mig}$	工序 $i$ 在其加工设备上的第 $g$ 个后工序
$N_{Qik}$	工序 $i$ 在其工序序列上的第 $k$ 个后续工序

**定义 1.** 工序序列: 内部工序彼此之间只存在串行关系的工序集合, 且集合中每道工序最多具有一个紧前工序和一个紧后工序.

设产品共有  $n$  道工序, 按加工开始时间顺序排列, 工序号为  $i (1 \leq i \leq n)$  综合调度问题满足以下约束条件:

$$T_{Si} \geq \max(T_{EF_{ij}}), \quad j \geq 1 \quad (1)$$

$$T_{Ei} = T_{Si} + T_i \quad (2)$$

$$T_{Ei} \leq \min(T_{SN_{if}}), \quad f \geq 1 \quad (3)$$

$$T_{Mi} \geq T_{M_{i-1}} \quad (4)$$

$$T_{Ti} = T_{T_{i1}}, T_{T_{i2}}, \dots, T_{T_{iJ}}, \quad J \geq 1 \quad (5)$$

$$T_{Si} \in T_{Ti} \quad (6)$$

问题的目标函数为

$$T = \min(\max(T_{Ei}) - \min(T_{Si})), \quad 1 \leq i \leq n \quad (7)$$

式 (1) 表示工序  $i$  的加工开始时间一定大于或等于工序  $i$  在工艺树中的紧前工序的加工结束时间; 式 (2) 表示工序  $i$  的加工结束时间等于其加工开始时间与其加工用时之和; 式 (3) 表示工序  $i$  的加工结束时间一定小于或等于其工艺树中所有紧后工序的最小加工开始时间; 式 (4) 表示工序  $i$  调度后所有已调度工序的最大加工完成时间一定大于或等于工序  $i - 1$  调度后所有已调度工序的最大加工完成时间; 式 (5) 表示工序  $i$  “准调度时间点” 集合由工序的  $J$  个“准调度时间点”组成; 式 (6) 表示工序  $i$  加工开始时间属于工序  $i$  “准调度时间点” 集合; 式 (7) 表示最晚加工结束工序的加工结束时间与最早加工开始工序的加工开始时间之差最小为单车间一般综合调度问题的调度目标.

## 2 算法设计思想

**定义 2.** 调度方案: 其中记录各个设备上目前已调度工序的信息.

**定义 3.** 工序调度方案: 以工序  $i$  (非最长工序序列上工序) 为最终调度工序的调度方案称为工序  $i$  调度方案, 记为  $D_i$ .

**定义 4.** 工序序列: 其中工序之间只存在串行关系的工序集合, 且每道工序在集合中最多具有一个紧前工序和一个紧后工序.

**定义 5.** 静态调度: 虚拟调度, 在调度工序时不考虑其他因素, 仅以其工艺树中紧前工序的加工结束时间作为当前调度工序的加工开始时间.

**定义 6.** 调度方案: 其中记录各个设备上目前已调度工序的信息.

在解决综合调度问题时, 根据加工工艺树特点可知, 调度工序须严格遵守其约定的偏序关系 (即串行工序之间的先后加工顺序), 除此之外可以任意调节工序的加工顺序, 使产品加工总用时尽可能小. 要达到这一目的, 往往是在遵守工艺树中约定偏序关系的前提下调整并行工序之间的加工顺序, 尽可能提高串行工序的紧密度和并行工序的并行性, 从而优化调度结果. 由于综合调度问题是 NP 问题, 工艺树中工序的加工偏序关系、加工用时和加工设备等参数不存在任何规律, 所以之前的研究, 在没有综合考虑工序及其相关工序 (前继或后续) 的加工用时、加工设备以及其相互间的偏序关系的前提下, 仅根据工序路径长度或设备忙等指标来决定工序的调度往往出现前言中所述的问题. 本文算法将在遵守工序间偏序关系的前提下, 结合工艺树自身特点为工序选择合适的加工时间, 同时提高串行工序的紧密度和并行工序并行性, 有效优化了综合调度算法, 下面举例说明所提算法设计思想.

设有产品加工工艺树如图 1 所示, 采用逆序调

度方式, 工艺树中的根节点无前继工序, 叶节点无后续工序. 工艺树中工序  $A_1, A_2, A_3, A_4$  和  $A_5$  组成的序列称为工序序列  $A$ , 工序  $B_1, B_2$  和  $B_3$  组成的序列称为工序序列  $B$ , 工序  $C_1$  和  $C_2$  组成的序列称为工序序列  $C$ , 每个工序序列中工序之间存在串行关系, 调度时将严格遵守约定的加工次序. 另外, 工序  $C_1$  与  $A_2$ , 工序  $B_1$  与  $A_1$  之间也存在着串行关系, 调度时将一并考虑. 如图 2 所示, 只考虑工序间的串行关系, 分别独立调度工序序列  $A, B$  和  $C$ , 得到图 2 中左侧所示 3 个调度甘特图, 将其叠加后得到图 2 中右侧所示的甘特图, 由于此时没有考虑并行工序之间的制约关系, 造成了并行工序加工时间重叠而形成冲突, 图 2 中  $a$  表示工序  $A_2$  和  $B_1$  发生冲突的时间区间;  $b$  表示工序  $A_4$  和  $B_3$  发生冲突的时间区间;  $c$  表示工序  $A_3$  和  $B_2$  发生冲突的时间区间. 由于发生冲突的工序为并行工序, 彼此之间没有加工偏序关系的约束, 其任意顺序的组合均满足工艺约束, 于是从工艺树自身特点出发, 寻找使加工总用时最小的并行工序组合方案.

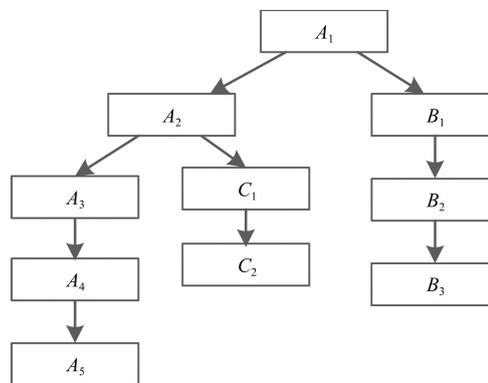


图 1 产品加工工艺树示例

Fig. 1 The sample of processing tree of product

以工序序列  $A$  与工序序列  $B$  相组合为例说明本文算法的设计思想. 首先以工序序列  $A$  的调度方案为基础, 在其上分别调度工序序列  $B$  中各工序, 寻找最优组合方案使所调度工序的加工总用时最小.

如图 3 所示, 以调度工序序列  $B$  中第 1 个工序  $B_1$  为例, 为了满足工艺树中的约束,  $B_1$  的最早加工开始时间为工序  $A_1$  的加工结束时间, 在此时间点之后的任一时间点都可做为工序  $B_1$  的加工开始时间, 但在工序  $B_1$  的加工设备上, 对实现调度目标有意义的加工时间点仅为时间点  $T_1$  和  $T_2$ , 因为这两个时间点分别代表了工序  $B_1$  与同设备并行工序不同组合顺序的最早加工开始时间. 在时间点  $T_1$  和时间点  $T_2$  分别试调度工序  $B_1$ , 得到两个工序  $B_1$  的试调度方案. 其中工序  $B_1$  在时间点  $T_1$  调度时, 影响了基础调度方案当中的相关工序, 需对这些工序进行调整

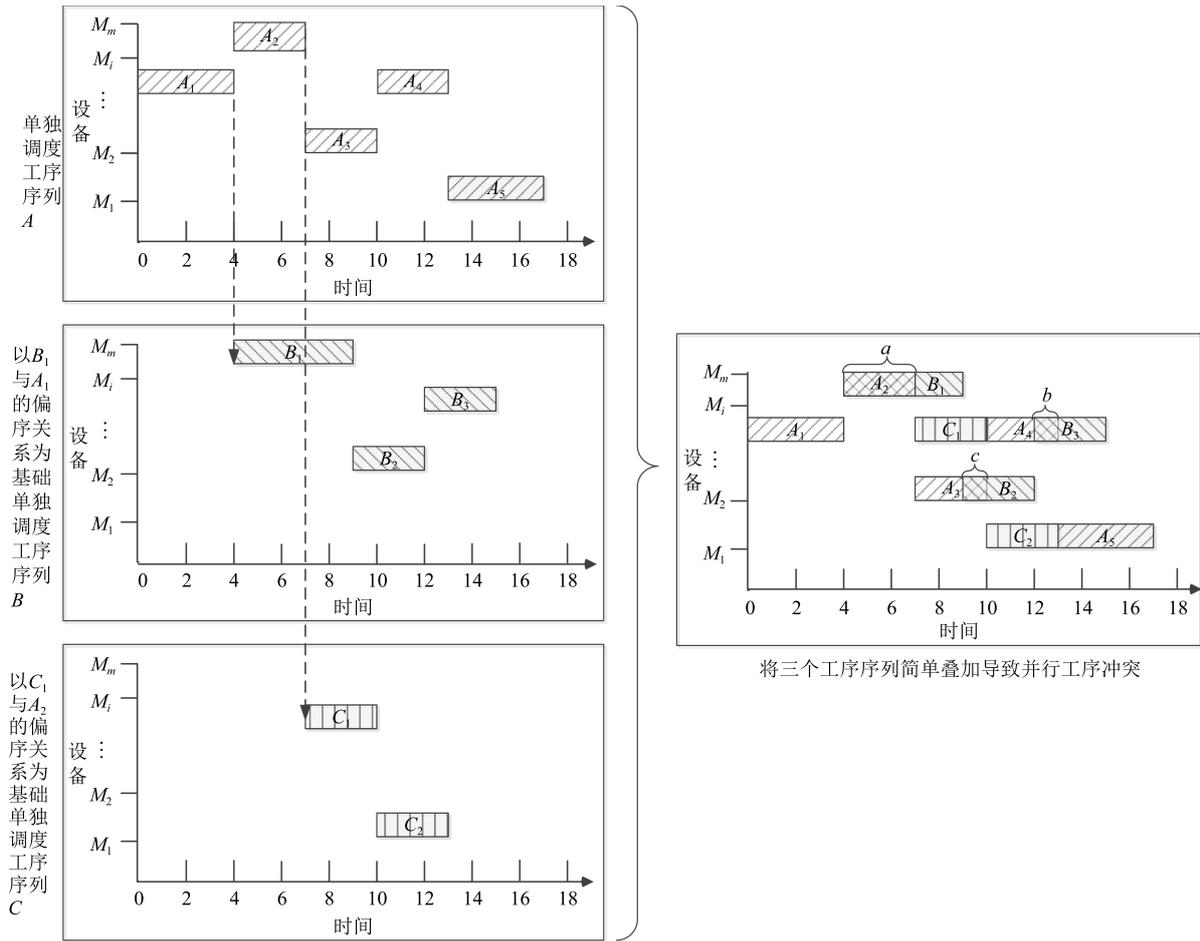


图 2 调度思想示意图 1

Fig. 2 Diagram 1 of the scheduling idea

调度再形成可行的工序  $B_1$  试调度方案. 计算 2 个试调度方案的加工总用时, 此时不会直接选择加工总用时少的方案作为工序  $B_1$  的调度方案, 还需考虑在每个试调度方案的前提下, 工序序列  $B$  中工序  $B_1$  所有后续工序最理想状态 (工序的加工开始时间即其紧前工序的加工结束时间) 下加工完成时间, 即静态调度工序  $B_1$  的所有后续工序所得加工完成时间, 比较此时间点与试调度方案的加工总用时, 取两时间点中较大者, 其代表在此时间点处调度工序可预见的最大加工用时. 例如, 图 3 中在时间点  $T_1$  处调度工序  $B_1$  所得试调度方案的加工总用时为 22, 静态调度工序  $B_1$  的所有后续工序所得加工完成时间为 15, 较大者为时间点 22; 在时间点  $T_2$  处调度工序  $B_1$  所得试调度方案的加工总用时为 17, 静态调度工序  $B_1$  的所有后续工序所得加工完成时间为 18, 较大者为时间点 18. 比较在两个时间点处调度工序  $B_1$  可预见的最大加工用时, 选择其中最小者所在的试调度方案做为工序  $B_1$  的调度方案. 本例具体为: 比较 22 和 18, 选择其中最小者 18 所在的试调度方

案, 即在时间点  $T_2$  处调度工序  $B_1$  的试调度方案做为工序  $B_1$  的调度方案. 其中, 静态调度的工序只作为选择方案的参考, 实际上此时并没有真正进行调度. 按照上述方法依次调度工序序列  $B$  上后续工序, 完成工序序列  $A$  与工序序列  $B$  的组合. 然后以工序序列  $A$  和  $B$  的组合方案为基础, 继续调度工序序列  $C$ , 直到工艺树中所有工序全部调度完毕.

### 3 策略分析与实现

**定义 7.** 初始调度方案: 只调度产品工艺树中最长工序序列上工序所形成的调度方案.

**定义 8.** 产品调度方案: 产品中的所有工序调度完成所形成的调度方案.

**定义 9.** 工序队列: 按工序的调度顺序存放工序的队列.

**定义 10.** 准调度时间点: 工序  $i$  在加工设备  $M_i$  上的若干备选加工开始时间点称为工序  $i$  的准调度时间点.

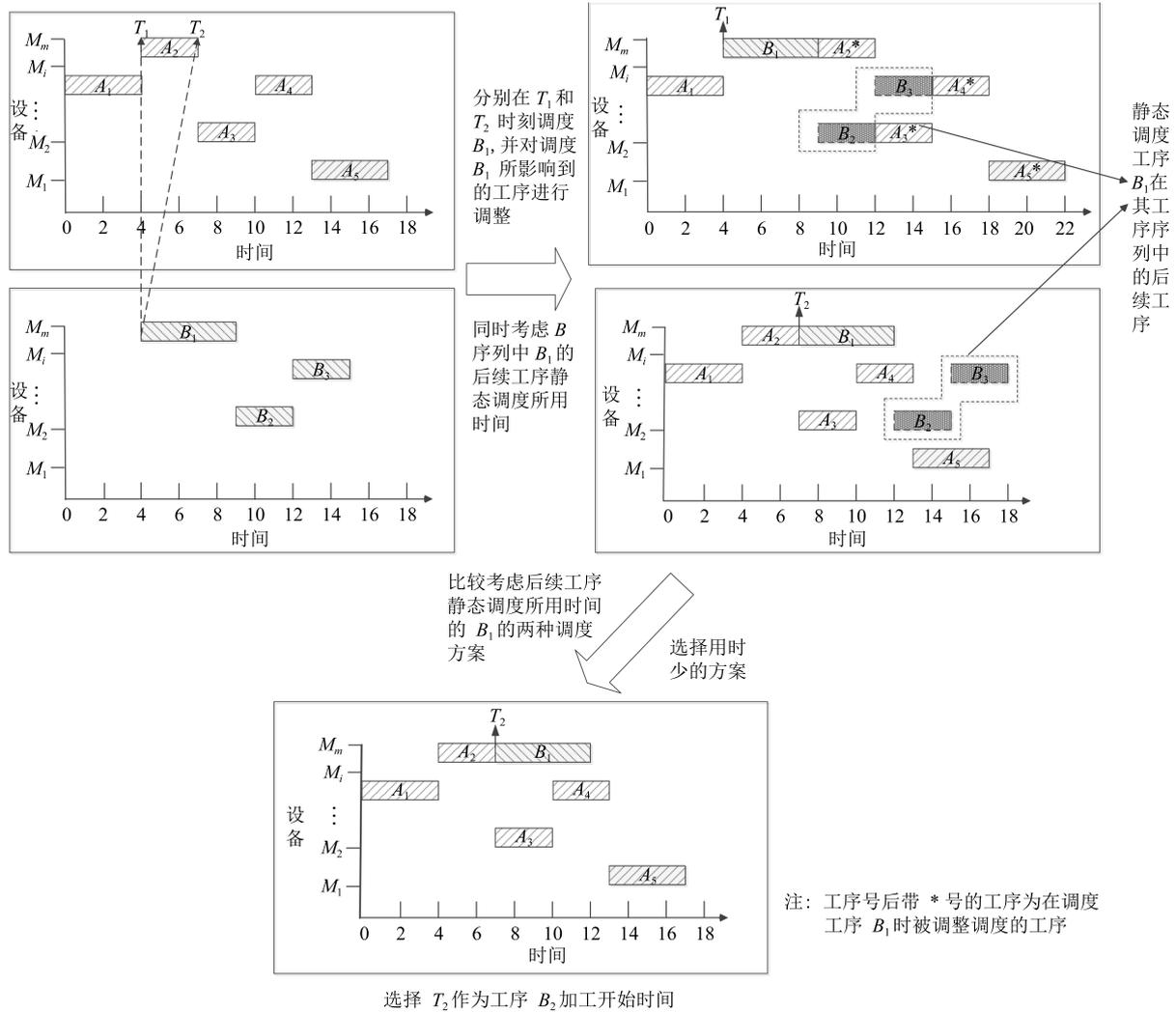


图 3 调度思想示意图 2

Fig. 3 Diagram 2 of the scheduling idea

**定义 11.** 调度方案链表: 按工序队列中工序的顺序存放初始调度方案及各工序调度方案的链表。

**定义 12.** 路径长度: 工序  $i$  与其在当前产品工艺树中所有前继工序的加工用时之和为工序  $i$  的路径长度。

**定义 13.** 最长工序序列: 其中所有工序加工用时之和最大的工序序列。

**定义 14.** 基础调度方案: 调度工序  $i$  (非最长工序序列上工序) 时, 要在其工序队列中紧前工序  $i-1$  的工序调度方案的基础上  $D_{i-1}$  进行调度, 那么方案  $D_{i-1}$  就是工序  $i$  的基础调度方案。

如图 4 所示, 设产品  $P$  共有  $n$  道工序, 即  $P_0, P_1, \dots, P_n, n = n_1 + n_2$ , 其中  $n_1$  为最工序序列上工序,  $n_2$  为非最长工序序列上所有工序. 所提算法将首先调度最长工序序列上工序, 生成初始调度方案, 然后利用考虑后续工序的择时策略分别调度工

序队列中剩余工序 (非最长工序序列上工序), 直至生成产品调度方案, 也就是产品  $P$  的生产计划。

### 3.1 逆序调度分析

择时策略的调度思想是在满足工序偏序关系的基础上, 为并行工序寻找一种能使当前所有已调度工序加工总用时最小的组合方式. 如图 5 所示, 工序  $i$  为调度工序, 工序  $F_{i1}$  为工序  $i$  在加工工艺树中加工结束时间最晚的紧前工序, 且此刻在其加工设备能够与其并行加工的工序按先后顺序为  $\beta_1, \beta_2, \dots, \beta_j$ , 共  $j$  个工序, 时间点  $T_1$  为工序  $F_{i1}$  的加工结束时间, 时间点  $T_2$  为工序  $\beta_j$  的加工结束时间, 工序  $i$  的加工开始时间  $T_{Si}$  应满足公式  $T_1 \leq T_{Si} \leq T_2$ .

在区间  $[T_1, T_2]$  上为工序  $i$  寻找使当前所有已调度工序加工总用时最小的时间点  $T_{Si}$  进行调度, 目

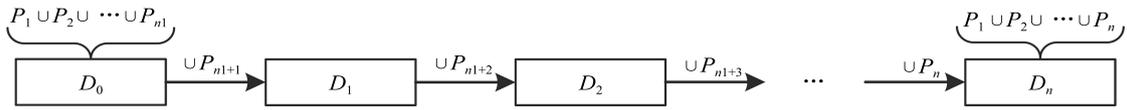


图4 产品P生产计划生成示意图

Fig. 4 The production plan generates schematic of product P

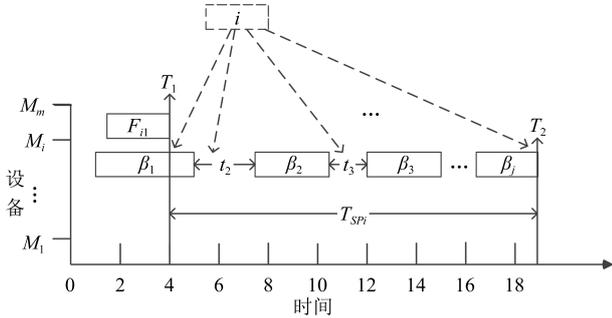


图5 工序i择时调度示意图

Fig. 5 The timing scheduling schematic of process i

前在区间  $[T_1, T_2]$  上已有并行工序  $\beta_1, \beta_2, \dots, \beta_j$  存在, 且工序  $i$  与其并行加工工序的组合共有  $j + 1$  种, 如下所示.

- 1)  $i, \beta_1, \beta_2, \dots, \beta_j$
- 2)  $\beta_1, i, \beta_2, \dots, \beta_j$
- 3)  $\beta_1, \beta_2, i, \dots, \beta_j$
- ⋮
- k)  $\beta_1, \beta_2, \dots, i, \beta_k, \dots, \beta_j$
- ⋮
- j)  $\beta_1, \beta_2, \dots, i, \beta_j$
- j + 1)  $\beta_1, \beta_2, \dots, \beta_j, i$

实现以上  $j + 1$  种组合的关键是确定工序  $i$  在每一种组合中的调度时间点. 这里并不是真正对工序  $i$  进行调度, 而是通过对工序  $i$  进行以上  $j + 1$  种试调度确定哪一种调度方案最合适. 所以每一种组合中工序  $i$  的加工开始时间称为工序  $i$  的“准调度时间点”. 确定工序  $i$  的“准调度时间点”, 可分为以下 3 种情况讨论.

**情况 1.** 第 1 种组合中工序  $i$  可选择的加工开始时间为工序  $F_{i1}$  的加工结束时间点, 即  $T_{Si} = T_{EF_{i1}}$ .

**情况 2.** 第  $k$  ( $2 \leq k \leq j$ ) 种组合中工序  $i$  加工开始时间  $T_{Si} \in t_k$  ( $2 \leq k \leq j$ ), 设  $T \in t_k$ , 为了提高串行工序紧密度及其后续工序之间的并行性, 在每种组合中, 都将选择使工序  $i$  尽早加工的时间点, 使其满足  $\min(T - T_{E\beta_{k-1}})$ , 如图 4 所示, 工序  $i$  可选择的时间点为  $t_k$  区间的下界, 即  $T = [t_k] = T_{E\beta_{k-1}}$ , 工序  $i$  的加工开始时间为  $T_{Si} = T_{E\beta_{k-1}}$ .

**情况 3.** 第  $j + 1$  种组合中工序  $i$  可选择的加工开始时间为工序  $\beta_j$  的加工开始时间点, 即  $T_{Si} = T_{E\beta_j}$ .

以上 3 种情况中, 情况 2 和情况 3 中工序  $i$  的加工开始时间  $T_{Si}$  由其在组合中的紧前工序  $\beta_k$  ( $1 \leq k \leq j$ ) 的加工结束时间  $T_{E\beta_k}$  来决定, 由于调度工序  $i$  时, 工序  $\beta_1, \beta_2, \dots, \beta_j$  已被调度, 所以  $T_{E\beta_k}$  是确定值, 直接使用即可, 不需要讨论.

情况 1 中工序  $i$  的加工开始时间由其工艺树中最晚结束紧前工序的加工结束时间  $T_{EF_{i1}}$  决定. 由于加工工艺树的结构特点, 不是其中所有节点都具有紧前工序, 下面分别从正序调度 (即按从叶节点向根节点方向调度) 和逆序调度 (即从根节点向叶节点方向调度) 两个角度进行分析.

### 3.1.1 按正序调度工序 i

1) 当工序  $i$  为叶节点. 此时工序  $i$  不存在紧前工序, 若  $i$  为工艺树中首个被调度工序, 其加工开始时间  $T_{Si} = 0$ ; 若  $i$  为非首个被调度工序, 则较难确定其加工开始时间;

2) 当工序  $i$  非叶节点. 此时工序  $i$  一定存在紧前工序, 且可能不唯一, 需对工序  $i$  的  $n$  个紧前工序的加工结束时间进行比较, 选择其中最大者作为工序  $i$  的加工开始时间, 即  $T_{Si} = \max(T_{EF_{ij}})$  ( $1 \leq j \leq n$ ).

### 3.1.2 按逆序调度工序 i

1) 当工序  $i$  为根节点. 按第 3.2 节中所述, 此时工序  $i$  一定为工艺树中首个调度的工序, 所以工序  $i$  的加工开始时间  $T_{Si} = 0$ .

2) 当工序  $i$  为非根节点. 此时根据加工工艺树的结构特点可知, 工序  $i$  一定有且仅有唯一紧前工序, 其加工开始时间为其紧前工序的加工结束时间, 即  $T_{Si} = T_{EF_{i1}}$ . 综上所述, 逆序调度更易于本文算法实现.

## 3.2 工序序列排序策略分析与实现

工序序列排序分为工序序列内部工序排序和工序序列间排序两部分.

### 3.2.1 工序序列内部工序排序分析

设有工序序列  $\alpha = \alpha_i \prec \alpha_{i+1}$  ( $1 \leq i \leq n$ ). 通过上述可知存在  $T_{T\alpha_{i+1}} = T_{E\alpha_i}$ , 即工序  $\alpha_{i+1}$  的第 1 个“准调度时间点”由工序  $\alpha_i$  (即  $\alpha_{i+1}$  在加工工

艺树中的紧前工序) 的加工结束时间确定. 调度工序  $\alpha_i$  是调度工序  $\alpha_{i+1}$  的基础, 所以每个工序序列内部工序的调度顺序为  $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$ , 即以逆序工艺树中工序偏序关系为依据排序. 以图 6 中工序序列 (a) 为例, 工序序列内部排序顺序为:  $W_1 \rightarrow W_2 \rightarrow W_3 \rightarrow W_4 \rightarrow W_5$ .

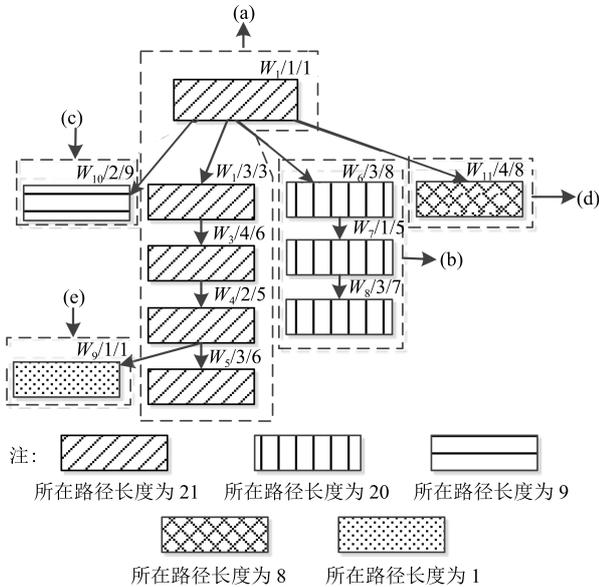


图 6 工序序列排序策略序列划分示意图

Fig. 6 The schematic of sequence divided by operation sequence sorting strategy

### 3.2.2 工序序列间排序分析

工序序列间排序分为两步: 1) 确定工序序列, 即将工艺树中的所有工序划分成若干工序序列; 2) 对工序序列排序, 即确定工序序列加工的先后关系.

首先, 择时策略调度工序时需已知该工序在工艺树中的紧前工序加工结束时间, 所以要保证调度工序时, 其工艺树中紧前工序已被调度, 即先加工的工序序列中必须包含后加工工序序列中工序的紧前工序. 以图 6 中工艺树为例, 工序序列间排序顺序为: (a)  $\rightarrow$  (b)  $\rightarrow$  (c)  $\rightarrow$  (d)  $\rightarrow$  (e).

工序序列排序策略实现时将序列内部排序和序列间排序一并处理, 即划分工序序列时, 先确定的序列排在前面, 工序序列确定后, 随即对其内部工序按加工偏序排序, 再确定下一个工序序列, 依次类推, 直到所有工序序列排序完毕. 以图 6 中工艺树为例, 经过工序序列排序策略排序后, 其中工序的调度次序为:  $W_1 \rightarrow W_2 \rightarrow W_3 \rightarrow W_4 \rightarrow W_5 \rightarrow W_6 \rightarrow W_7 \rightarrow W_8 \rightarrow W_{10} \rightarrow W_{11} \rightarrow W_9$ .

### 3.2.3 工序序列策略算法实现

工序序列排序策略算法实现步骤如下:

步骤 1.  $i = 1$ ;

步骤 2. 设有工序队列  $Qu$ ;

步骤 3. 判断加工工艺树 (或森林) 是否空, 不为空转至步骤 4, 为空转至步骤 12;

步骤 4. 计算当前工艺树中每个叶子节点的路径长度;

步骤 5. 选路径长度最大的叶子节点, 若唯一转至步骤 7, 若不唯一转至步骤 6;

步骤 6. 选择其路径上工序最多的叶子节点;

步骤 7. 将其和其所在路径上所有节点按偏序关系由前到后的顺序确定为第  $i$  工序序列 (第 1 个被调度的工序序列);

步骤 8. 将第  $i$  工序序列中工序, 依次入工序队列  $Qu$ ;

步骤 9. 将第  $i$  工序序列中工序在加工工艺树 (或森林) 中删除;

步骤 10.  $i + 1$ ;

步骤 11. 转至步骤 3;

步骤 12. 返回工序队列  $Qu$ , 结束.

## 3.3 择时调度策略分析与实现

### 3.3.1 择时策略所涉及工序的确定问题分析

在图 5 中, 工序  $i$  的加工开始时间  $T_{Si}$  满足  $T_{Si} \in [T_1, T_2]$ , 时间点  $T_2$  是当前在工序  $i$  加工设备  $M_i$  上已调度的最后一个工序  $\beta_j$  的加工结束时间  $T_{E\beta_j}$ ; 时间点  $T_1$  是工序  $i$  在工艺树中的紧前工序加工结束时间, 这一时间点及设备  $M_i$  上的工序调度情况有 3 种, 如图 7 所示.

情况 1 如图 7(a) 所示,  $T_{Ti1} \geq T_{E\beta}$  成立, 此时工序  $i$  的调度并不会影响工序  $\beta$ , 工序  $\beta$  将不作为与工序  $i$  并行的工序与其进行组合, 在试调度  $i$  时无须考虑工序  $\beta$ ; 情况 2 如图 7(b) 所示,  $T_{S\beta} < T_{Ti1} < T_{E\beta}$  成立, 可见工序  $i$  与工序  $\beta$  之间存在并行关系, 在时间点  $T_{Ti1}$  调度工序  $i$  时一定会与工序  $\beta$  发生冲突, 此时需将工序  $\beta$  调整到工序  $i$  之后, 即实现组合  $i, \beta, \dots$ ; 同时由于公式  $T_{Ti2} = T_{E\beta}$  成立, 可实现组合  $\beta, i, \dots$ ; 情况 3 如图 7(c) 所示,  $T_{Ti1} \leq T_{S\beta}$  成立, 此时工序  $i$  与工序  $\beta$  之间存在并行关系, 在时间点  $T_{Ti1}$  调度工序  $i$  时可能会与工序  $\beta$  发生冲突, 此时需将工序  $\beta$  调整到工序  $i$  之后, 即实现组合  $i, \beta, \dots$ ; 同时由于  $T_{Ti2} = T_{E\beta}$  成立, 可实现组合  $\beta, i, \dots$ . 综上所述, 情况 2 和情况 3 需在时间点  $T_{Ti1}$  调度工序  $i$ , 同时需对影响到的工序  $\beta$  进行调整, 具体方法在第 3.3.2 节详细介绍.

### 3.3.2 试调度工序后引起的工序冲突问题

如图 8 所示, 调度工序  $i$ , 工序  $F_i$  为其在加工工艺树中的紧前工序, 工序  $\beta_1, \beta_2, \beta_3$  为与工序  $i$  同设备的并行工序, 工序  $A$  是工序  $\beta_1$  在工艺树中的紧后工序, 工序  $B$  是工序  $A$  在工艺树中的紧后工序, 图

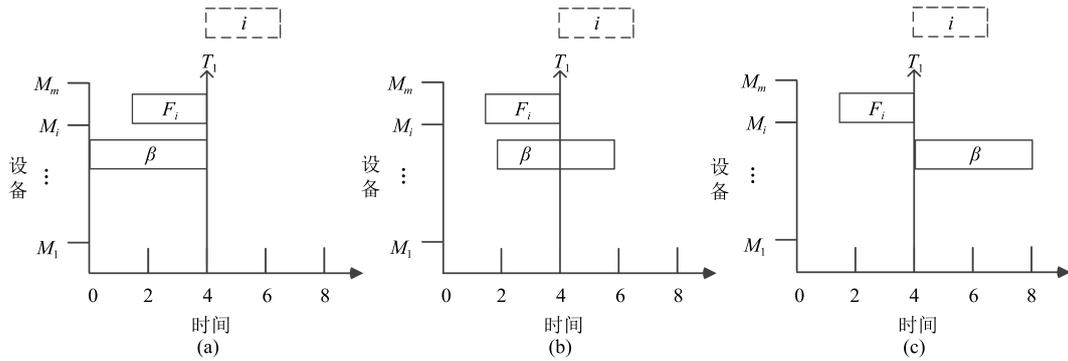


图 7 时间点  $T_1$  所涉及已调度工序

Fig. 7 The scheduled process associated with the  $T_1$

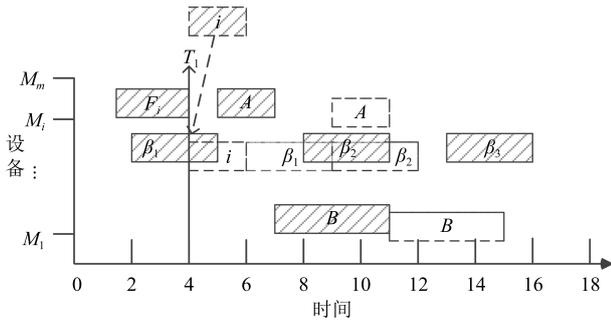


图 8 处理工序  $i$  试调度后工序发生冲突示意图

Fig. 8 The diagram of process conflict when process  $i$  tried to scheduled

中斜线填充的工序为调度工序  $i$  时已调度的工序, 白色填充虚线框的工序为在时间点  $T_1$  处调度工序  $i$  后被调整的工序状态. 在各个“准调度时间点”试调度工序  $i$  时, 往往会与已调度工序发生冲突. 例如, 工序  $i$  在时间点  $T_{T1} = 4$  进行试调度, 此时会与其在加工设备上的并行工序  $\beta_1$  在时间段  $[4, 5]$  发生冲突, 解决办法是在时间点  $T_{T1}$  调度工序  $i$ , 将工序  $\beta_1$  调整到工序  $i$  之后调度, 即  $T_{S\beta_1} = T_{Ei} = 6$ , 调度工序  $\beta_1$ , 得到  $(T_{E\beta_1} = 9) > (T_{SA} = 5)$ , 于是工序  $\beta_1$  与其在工艺树中的紧后工序  $A$  相冲突, 调整工序  $A$  的加工开始时间, 令  $T_{SA} = 9$ , 与此同时, 得到  $(T_{E\beta_1} = 9) > (T_{S\beta_2} = 8)$ , 于是工序  $\beta_1$  与其设备上的紧后工序  $\beta_2$  相冲突, 调整工序  $\beta_2$  的加工开始时间, 令  $T_{S\beta_2} = 9$ , 继而, 新调整的工序  $A$  与工序  $B$  发生冲突, 调整工序  $B$ , 令  $T_{SB} = 11$ .

综上所述, 对工序试调度可能引起该工序与其加工设备上已调度的紧后工序发生冲突, 调整工序时又可能引起两类冲突, 即: 该工序与其加工设备上已调度的紧后工序发生冲突和该工序与其在加工工艺树中已调度的紧后工序发生冲突. 继而调整工序, 往往又会影响到其他工序, 依次类推. 在解决冲突时, 需对试调度工序和所有被调整的工序逐一排查, 直到冲突全部解决.

### 3.3.3 择时调度策略算法实现

设当前调度工序为  $i$ , 其工序序列中的紧前工序为  $i - 1$ , 则基础调度方案为  $D_{i-1}$ , 工序  $i$  加工设备为  $M_i$ , 工序  $P_i$  加工工艺树中紧前工序为  $F_{i1}$ , 其第 1 个准调度时间点确定为  $T_{EFi1}$ , 在  $M_i$  上时间点  $T_{EFi1}$  之后共有  $n$  个工序加工结束时间点.

择时调度策略算法实现步骤如下:

步骤 1.  $j = 1, k = 1$ ;

步骤 2. 将  $T_{EFi1}$  作为工序  $P_i$  的第 1 个“准调度时间点”;

步骤 3. 判断在设备  $M_i$  上时间点  $T_{EFi1}$  是否为空闲时间或某工序加工结束时间点, 若是则转至步骤 7, 否则转至步骤 4;

步骤 4. 设备  $M_i$  上时间点  $T_{EFi1}$  处正在加工工序设为工序  $W$ , 令  $T_{SW} = T_{Ei}$ , 转至步骤 7;

步骤 5. 判断  $k \leq n$  是否成立, 成立转至步骤 6, 否则转至步骤 27;

步骤 6. 在设备  $M_i$  上以时间点  $T_{EFi1}$  为起点, 找到第  $k$  个工序加工结束时间点, 作为工序的第  $j$  个“准调度时间点”;

步骤 7. 在第  $j$  个“准调度时间点”上调度工序  $i$ ;

步骤 8. 设有工序队列  $adjust\_Q$ ;

步骤 9. 将工序  $i$  入队  $adjust\_Q$ ;

步骤 10. 判断工序队列  $adjust\_Q$  是否为空, 为空转至步骤 24, 否则转至步骤 11;

步骤 11. 出队操作, 取出工序设为  $A$ ;

步骤 12. 判断工序  $A$  在设备  $M_A$  上是否存在紧后工序  $N_{MA1}$ , 是则转至步骤 13, 否则转至步骤 16;

步骤 13. 判断工序  $N_{MA1}$  的加工开始时间是否小于工序  $A$  的加工结束时间, 是则转至步骤 14, 否则转至步骤 16;

步骤 14. 令  $T_{SN_{MA1}} = T_{EA}$ ;

步骤 15. 将工序  $N_{MA1}$  入队  $adjust\_Q$ ;

**步骤 16.** 判断工序  $A$  在加工工艺树中是否存在已调度紧后工序, 是则转至步骤 17, 否则转至步骤 10;

**步骤 17.** 设工序  $A$  在加工工艺树中的已调度紧后工序数为  $m$ ;

**步骤 18.** 设  $t = 1$ ;

**步骤 19.** 判断  $t \leq m$  是否成立, 是则转至步骤 20, 否则转至步骤 10;

**步骤 20.** 判断工序  $N_{At}$  的加工开始时间是否小于工序  $A$  的加工结束时间, 是则转至步骤 21, 否则转至步骤 23;

**步骤 21.** 令  $T_{N_{At}} = T_{EA}$ ;

**步骤 22.** 将工序  $N_{At}$  入队列  $adjust\_Q$ ;

**步骤 23.**  $t++$ , 转至步骤 19;

**步骤 24.** 生成第  $j$  个工序  $i$  试调度方案  $D_{ij}$ ;

**步骤 25.**  $k++$ ,  $j++$ ;

**步骤 26.** 转至步骤 5;

**步骤 27.** 结束.

### 3.4 考虑后续工序策略分析与实现

#### 3.4.1 考虑后续工序策略分析

择时策略在若干“准调度时间点”试调度工序, 目的是找到一个使当前所有已调度工序加工总用时最小的时间点作为调度工序的加工开始时间, 如图 9 所示, 加工工艺树中工序序列 1 由工序  $A_1, A_2, A_3, A_4$  和  $A_5$  组成; 工序序列 2 由工序  $B_1, B_2$  和  $B_3$  组成. 按择时调度策略调度工序  $B_1$  时, 工序  $B_1$  的“准调度时间点”分别为 4, 9 和 17, 选择使当前所有已调度工序加工总用时最小的时间点  $T_1 = 9$  作为调度工序  $B_1$  的加工开始时间, 并在此基础上调度其后续工序  $B_2$  和  $B_3$ . 产品调度甘特图如图 10(a) 图所示, 产品加工总用时为 23; 相反, 如果选择时间点  $T_1 = 4$  作为调度工序  $B_1$  的加工开始时间, 并在

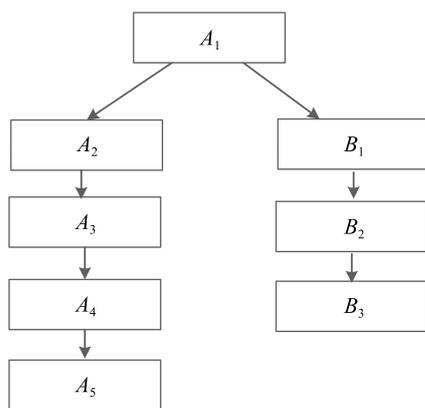


图 9 产品工艺树

Fig. 9 The processing tree of product

此基础上调度其后续工序  $B_2$  和  $B_3$ . 产品调度甘特图如图 10(b) 图所示, 产品加工总用时为 21.

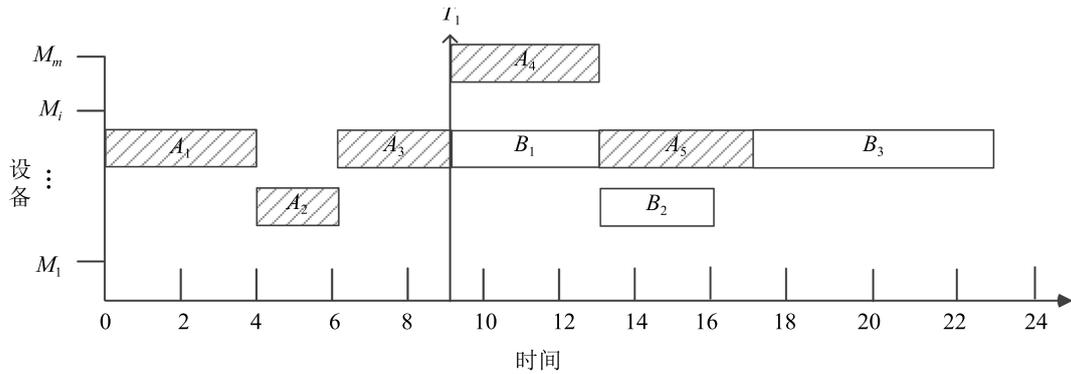
可见, 仅靠择时调度策略确定工序调度时间点并不一定比其他落选时间点的调度结果更好, 原因是: 择时策略在选择工序调度时间点时依据是使当前已调度工序加工总用时最小, 这并不等于使产品中所有工序加工总用时最小. 择时策略本质上是以使当前已调度工序的加工总用时最小为目标的最佳适应<sup>[8]</sup> 调度方法, 其并没有考虑当前未被调度的工序. 此时未被调度工序分为两类: 与该调度工序并行的工序 (即未调度的其他工序序列上工序) 和与该调度工序串行的工序 (该工序与在工序序列上的后续工序). 前一类工序由于与该工序并行, 彼此之间不存在制约关系, 不予考虑; 下面讨论后一类工序的调度.

当前调度工序的加工结束时间制约着其后续串行工序的加工开始时间, 若当调度工序加工结束时间较晚, 会使其后续串行工序加工开始时间延后, 导致工序间并行性差, 影响产品调度结果, 所以需在调度当前工序时, 同时考虑其对同工序序列上后续工序的影响. 当前有两种考虑同工序序列上后工序的方法, 下面分别阐述.

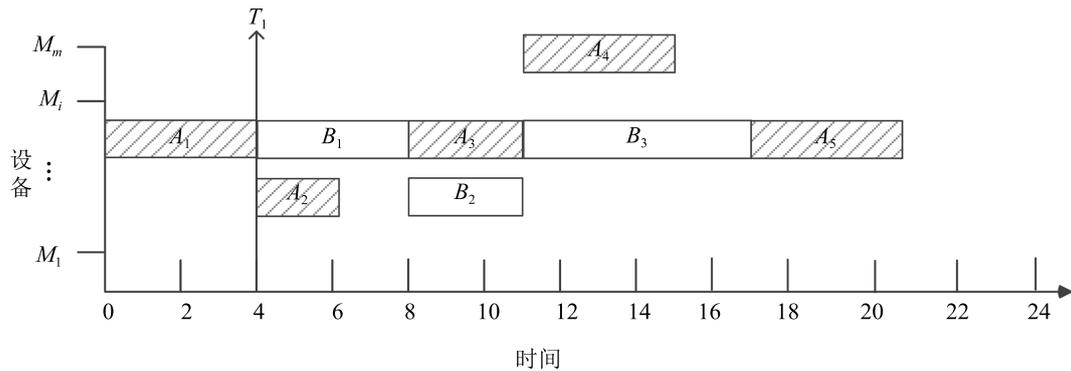
设当前调度工序为  $B_1$ , 其工序序列上的后续工序为  $B_2$ , 设工序  $B_1$  在加工设备上的“准调度时间点”个数分别为  $N$  ( $N \geq 1$ ) 个,  $B_1$  在第  $J$  ( $1 \leq j \leq N$ ) 个“准调度时间点”上调度后, 工序  $B_2$  的“准调度时间点”为  $K_j$  ( $1 \leq j \leq N$ ) 个.

**方法 1.** 将工序  $B_1$  及其后续工序  $B_2$  在每个“准调度时间点”穷举调度, 选择所有试调度方案中加工总用时最小方案, 其中的工序  $B_1$  所选择调度时间点即为工序  $B_1$  的理想调度时间点. 该方法首先在工序  $B_1$  的  $N$  个“准调度时间点”上进行  $N$  次择时操作, 得到工序  $B_1$  的  $N$  个试调度方案; 然后在每一个工序  $B_1$  试调度方案上的  $K_j$  ( $1 \leq j \leq N$ ) 个“准调度时间点”上分别试调度工序  $B_2$ , 所做择时操作为  $K_1 + K_2 + \dots + K_N$  次, 生成  $B_2$  试调度方案  $K_1 + K_2 + \dots + K_N$  个, 然后在其中选择已调度工序加工总用时最少的方案. 在调度工序  $B_1$  和  $B_2$  的过程中所做择时操作次数共为  $K_1 + K_2 + \dots + K_N$  次. 且该例中工序  $B_1$  的后续工序数为 1, 随着其后续工序数的增加, 择时操作次数将呈指数增长, 使算法复杂度陡增, 显然不利于算法的应用.

**方法 2.** 以工序  $B_1$  的每一个试调度加工结束时间为起点静态调度其工序序列上后续工序, 其工序序列上最后一道工序的加工结束时间记为  $TR$ , 是理想状态下工序  $B_1$  所在工序序列加工结束时间, 设其工序序列上最后一道工序的动态调度加工结束时间为  $TL$ ,  $TR \leq TL$  成立; 试调度工序  $B_1$  后所得工



(a) 在  $T_1 = 9$  时间点调度工序  $B_1$  时所得方案  
 (a) The scheme of scheduling process  $B_1$  at the time  $T_1 = 9$



(b) 在  $T_1 = 4$  时间点调度工序  $B_1$  时所得方案  
 (b) The scheme of scheduling process  $B_1$  at the time  $T_1 = 4$

图 10 仅使用择时策略在不同时间点调度工序  $B_1$  方案比较

Fig. 10 Scheme comparison of Scheduling  $B_1$  in different time when only use timing strategy

序  $B_1$  试调度方案的加工总用时记为  $T_{MB_1}$ , 计算  $\max(T_{MB_1}, TR)$ , 其值记为  $T_p$ , 是试调度工序  $B_1$  后当前可预见的产品最少加工用时, 设产品加工总用时为  $TP$ ,  $T_p \leq TP$  成立.

例如产品工艺树为图 9 所示工艺树, 如图 11 (a) 所示, 当前调度工序为工序  $B_1$ , 其选择的准调度时间点为  $T_{TB12} = 9$ , 调度工序  $B_1$  后所得  $T_{TB1} = 17$ , 以工序  $B_1$  的加工结束时间  $T_{TB1} = 13$  为起点静态调度工序  $B_2$  和  $B_3$ , 所得  $TR = 22$ , 计算  $\max(T_{MB12}, TR) = 22$ ; 如图 11 (b) 所示, 当前调度工序为工序  $B_1$ , 其选择的准调度时间点为  $T_{TB11} = 4$ , 调度工序  $B_1$  后所得  $T_{MB1} = 19$ , 以工序  $B_1$  的加工结束时间  $T_{EB1} = 8$  为起点静态调度工序  $B_2$  和  $B_3$ , 所得  $TR = 17$ , 计算  $\max(T_{MB11}, TR) = 19$ ; 计算  $\min(\max(T_{MB11}, TR), \max(T_{MB12}, TR)) = 19$ , 显然当工序  $B_1$  在  $T_{TB11} = 4$  调度时, 当前可预见的产品最小加工用时最小.

### 3.4.2 考虑后续工序策略实现

设调度工序为  $i$ , 工序  $i$  在其工序序列上的后续

工序共有  $n$  个, 当前择时调度策略算法所得某试调度方案为  $D_{ik}$ .

考虑后续工序策略算法实现步骤如下:

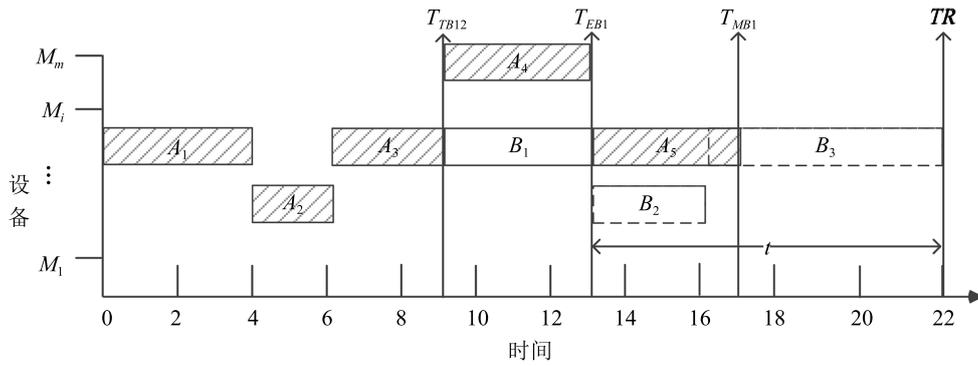
- 步骤 1.  $j = 1, ST_{ik} = T_{Ei}$ ;
- 步骤 2. 判断  $j \leq n$  是否成立, 成立则转至步骤 3, 否则转至步骤 4;
- 步骤 3.  $ST_{ik} + T_{N_{Qij}}$ , 转至步骤 2;
- 步骤 4. 返回  $ST_{ik}$ , 结束.

## 4 算法设计及复杂度分析

### 4.1 算法设计

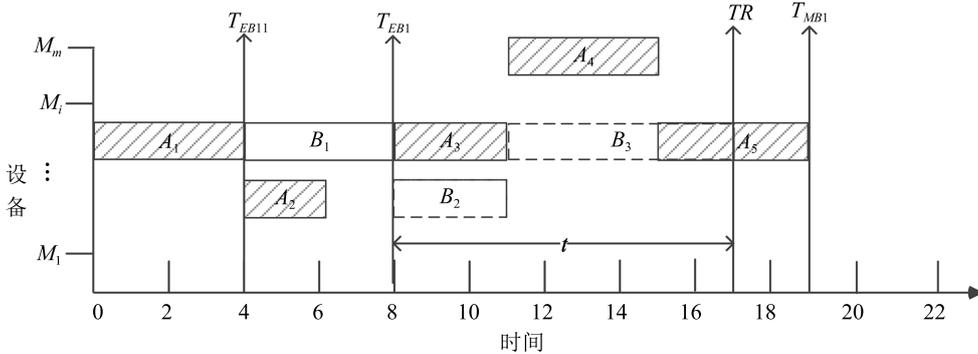
本文算法实现步骤如下:

- 步骤 1. 对产品加工工艺树中各工序间偏序关系一一取反, 得到逆序产品加工工艺树;
- 步骤 2. 应用工序序列排序策略为各工序排出调度顺序, 结果存入工序队列  $Qu$ ;
- 步骤 3. 将最长工序序列上工序依次出队, 调度其形成初始调度方案  $D_0$ ;
- 步骤 4. 判断  $Qu$  是否为空, 为空则转至步骤



(a) 调度工序  $B_1$  后形成的已调度工序加工总用时最小的方案

(a) The scheme that has the minimum amount of time to process the processes after the scheduling process  $B_1$



(b) 调度工序  $B_1$  同时考虑后续工序所得调度方案

(b) The scheme of scheduling the process  $B_1$  when considering posterior process

图 11 采用考虑后续工序策略在不同时间点调度工序  $B_1$  方案比较

Fig. 11 The comparison of scheme of scheduling  $B_1$  in different time

17, 否则转至步骤 5;

**步骤 5.** 对  $Qu$  做出队操作, 取得当前调度工序  $i$ ;

**步骤 6.** 对工序  $i$  应用择时调度策略, 得到  $n_i$  个试调度方案  $D_{ik}$  ( $1 \leq k \leq n_i$ );

**步骤 7.**  $k = 1, m = 0, p = NULL, j$ ;

**步骤 8.** 判断  $k \leq n_i$  是否成立, 是则转至步骤 9, 否则转至步骤 16;

**步骤 9.** 以试调度方案  $D_{ik}$  为基础, 应用考虑后续工序策略, 得到工序  $i$  后续工序静态调度完成时间  $ST_{ik}$ ;

**步骤 10.** 设  $D_{ik}$  的加工总用时为  $D_{ik}.totaltime$ , 计算

$$R_{ik} = \max(ST_{ik}, D_{ik}.totaltime)$$

**步骤 11.** 判断  $k = 1$  是否成立, 是则转至步骤 12, 否则转至步骤 13;

**步骤 12.**  $m = R_{ik}, j = k$ , 转至步骤 15;

**步骤 13.** 判断  $R_{ik} < m$  是否成立, 成立则转至步骤 14, 否则转至步骤 15;

**步骤 14.** 令  $m = R_{ik}, j = k$ ;

**步骤 15.**  $k++$ , 转至步骤 8;

**步骤 16.** 将试调度方案  $D_{ij}$  确定为工序  $P_i$  的调度方案  $D_i$ , 转至步骤 4;

**步骤 17.** 形成产品调度甘特图并输出.

#### 4.2 算法复杂度分析

设产品工序总数为  $N$ , 最长工序序列上的工序数为  $n_1$ , 其他工序数为  $n_2$ , 等式  $N = n_1 + n_2$  成立, 产品加工设备数为  $M$ . 所提算法中共涉及到 3 个主要策略, 分别为: 工序序列排序策略、形成初始调度方案策略和考虑后续工序的择时策略. 3 个策略彼此串行执行, 其中复杂度最大者将作为本算法的复杂度.

1) 工序序列排序策略. 根据工序序列排序策略, 排序分为工序序列间排序和工序序列内部工序排序, 算法实现时将这两种排序同步进行. 设工艺树叶节点的个数为  $nl$ , 工序序列间排序时, 第 1 步计算当前存在叶子节点的路径长度操作, 此时所做的累加运算次数最多为  $k$  ( $0 \leq k \leq N - nl$ ) 次, 其中  $nl$  最小为 1, 即  $k$  最大可取  $N - 1$ ; 第 2 步比较当前叶子节点的路径长度, 选择当前路径最长的叶子节点操作

最多需进行  $nl - 1$  次. 由工序序列排序策略所述, 由于有  $nl$  个叶子节点, 将生成  $nl$  个工序序列, 所以上述两步操作需要进行  $nl - 1$  次. 该策略的算法时间复杂度为  $(nl - 1) \times (N - 1 + nl - 1)$ , 即  $O(N \times nl)$ .

2) 形成初始调度方案. 初始调度方案由最长工序序列上的  $n_1$  个工序调度形成, 调度时只需将每个工序的加工结束时间作为其紧后工序的加工开始时间即可, 此项操作的复杂度为  $O(n)$ .

3) 考虑后续工序的择时策略. 一台加工设备上最多可能加工  $N - M + 1$  个工序, 调度工序最坏的情况是, 该工序加工设备上已有  $N - M$  道工序, 此刻其调度的方案共有  $N - M + 1$  种, 且这些工序调度后都需要启动择时调整策略, 调整涉及工序队列中排在该工序之前的所有工序, 此时需进行  $N - 1$  次处理, 考虑后续工序静态调度的完成时间, 需将当前调度工序所在工序序列上未被调度的工序加工时间进行累加操作, 累加的次数分析如下: 当工序为工序队列中最后一道工序时, 其后续工序数为 0, 所以择时策略最多进行  $(N - M + 1) \times (N - 1 + 0)$  次处理. 此项操作的算法时间复杂度为  $O(N^2)$ . 由于上述分析是针对工艺树中非最长工序序列上的某一个工序, 实际调度需处理工序数为  $n_2$ . 综上所述, 该策略的算法时间复杂度为  $O(n_2 \times N^2)$ . 本文算法的时间复杂度取上述各项策略复杂度的最大值, 为  $O(n_2 \times N^2)$ .

### 5 实例分析

本文算法是理论分析的结果, 对综合调度问题具有普遍意义, 不以具体实例为依据, 为了方便读者理解算法, 下面通过一个实例进行分析对比.

某制造工厂单件产品订单记为产品 A, 由 33 道工序组成, 在 4 台设备上加工, 其产品工艺树如图 12 所示, 下面使用文献 [10]、文献 [12] 中算法和本文算法分别调度产品 A, 得到的调度甘特图如图 13~15 所示.

文献 [10] 采用动态关键路径的调度思想, 在调度产品 A 时, 每调度一个工序之前动态计算并选择出当前可调度工序中路径长度最长的工序, 若不唯一则选择用时最少的工序, 工序的加工次序为:  $A_{28}, A_{29}, A_{25}, A_{33}, A_{20}, A_{26}, A_{23}, A_{22}, A_{21}, A_{27}, A_{30}, A_{16}, A_{32}, A_{19}, A_{15}, A_{24}, A_{14}, A_{13}, A_{17}, A_{18}, A_7, A_9, A_{11}, A_{31}, A_8, A_6, A_{12}, A_{10}, A_5, A_4, A_3, A_2$  和  $A_1$ , 共用时为 23. 其调度效果之所以不理想, 是因为该算法只关注工序的路径长度, 而忽略了其加工设备, 加工用时以及其在工艺树中的相对位置等因素对调度结果的影响. 例如, 工序  $A_{24}$  因其路径长度 11 大于工序  $A_{17}$  路径长度 9, 被优先调度, 而工序  $A_{24}$  的加工用时为 4, 大于工序  $A_{17}$  的紧前工序  $A_{19}$  的加工用时 2, 工序  $A_{24}$  优先调度降低了工序  $A_{19}$  与工序  $A_{17}$  之间的紧密度; 工序  $A_7$  因其路径长度 8

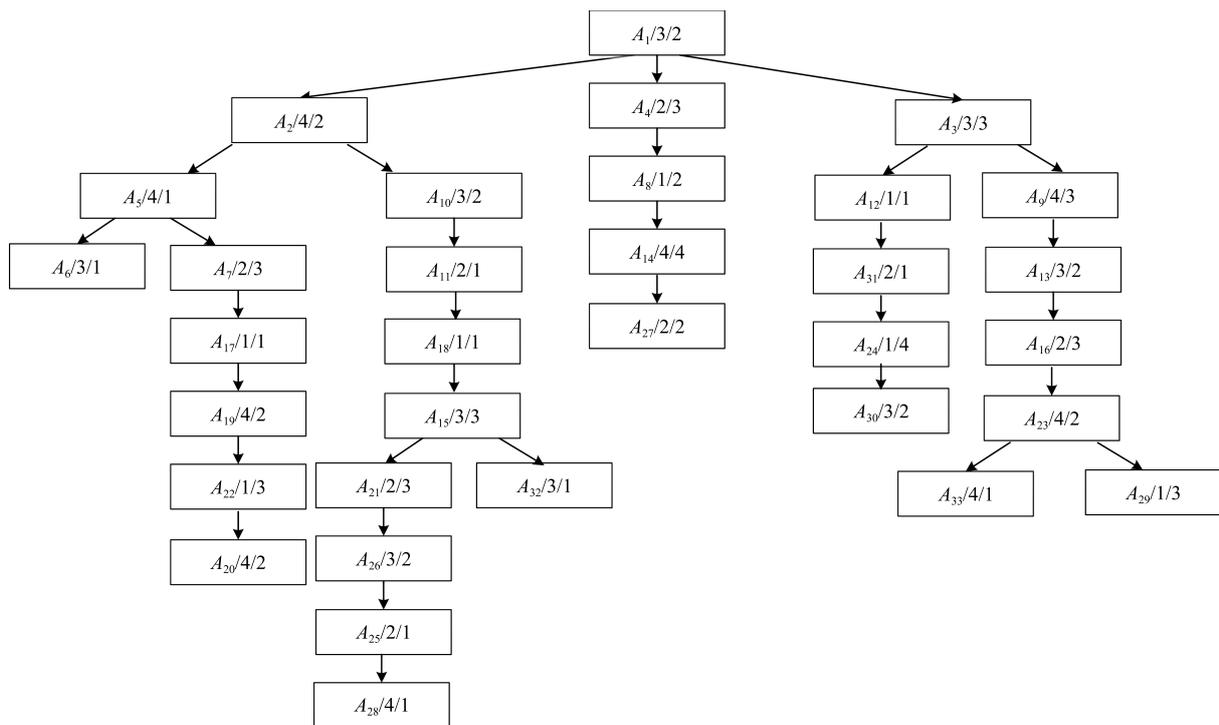


图 12 产品 A 的加工工艺树  
Fig. 12 The processing tree of product A

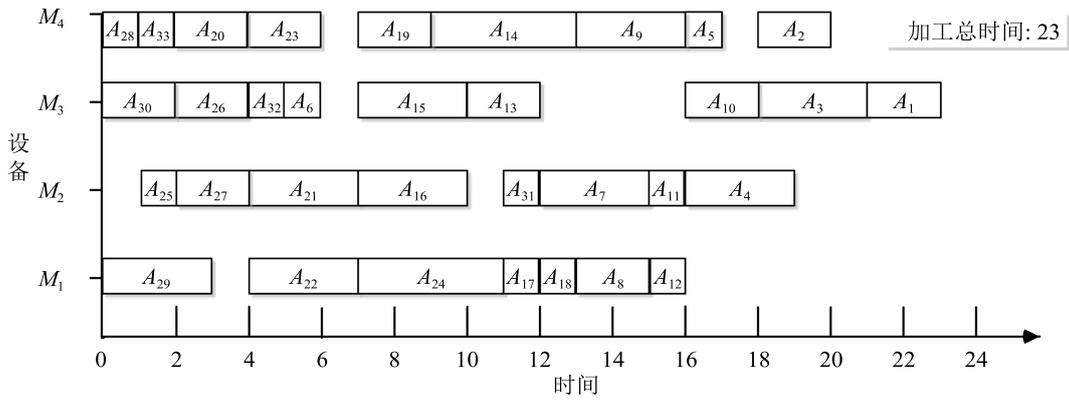


图 13 使用文献 [10] 算法所得甘特图

Fig. 13 Gantt chart using the algorithm of [10]

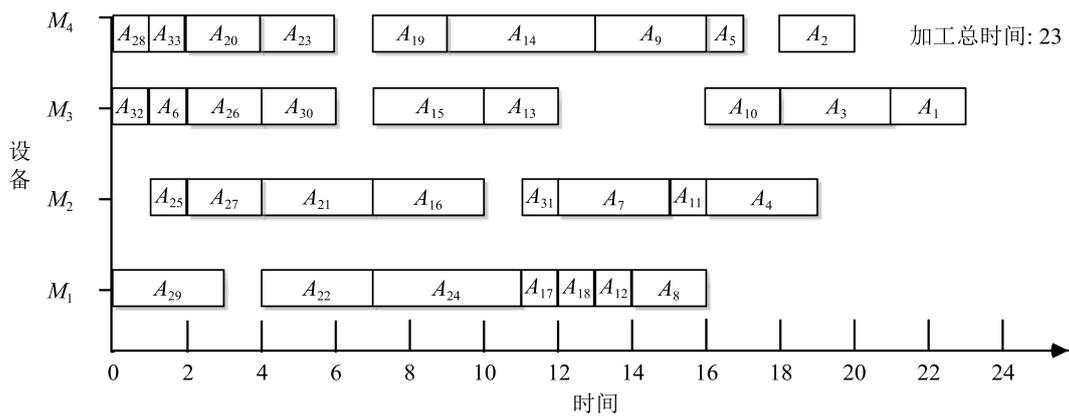


图 14 使用文献 [12] 算法所得甘特图

Fig. 14 Gantt chart using the algorithm of [12]

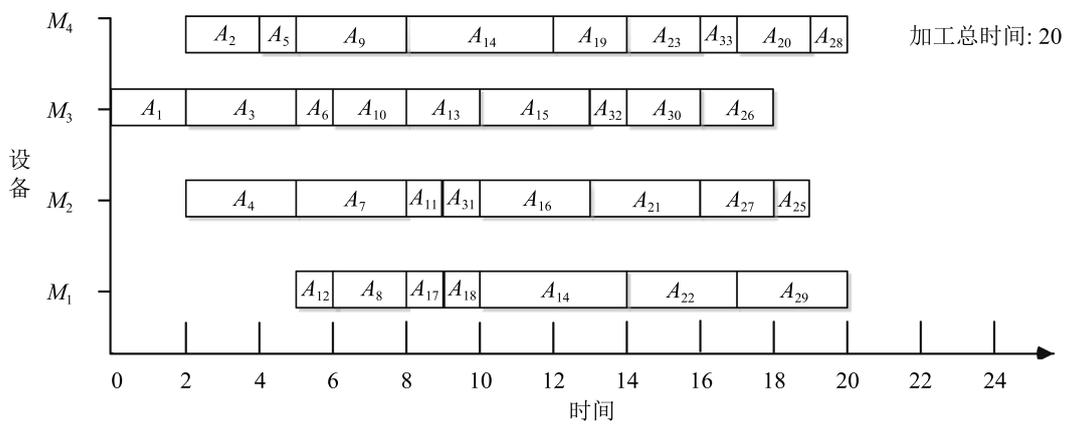


图 15 使用本文算法所得甘特图

Fig. 15 Gantt chart using the algorithm proposed

大于工序  $A_{11}$  路径长度 7, 被优先调度, 而工序  $A_7$  的加工用时为 3, 大于工序  $A_{11}$  的紧前工序  $A_{18}$  的加工用时 1, 工序  $A_7$  优先调度降低了工序  $A_{18}$  与工序  $A_{11}$  之间的紧密度。

文献 [12] 采用可回退抢占的设备驱动的调度思

想, 在调度产品  $A$  时, 以设备空闲事件驱动工序调度, 当同一时刻同一设备上可调度工序不唯一时, 选择父路径长的工序, 若设备驱动时刻, 新调度工序的加工备上有正在加工的工序, 则判断正在加工工序的已加工时间与其父路径长度之和是否小于新调度

工序的父路径长度, 若是, 则新调度工序将抢占已加工工序. 工序的加工次序为:  $A_{29}, A_{27}, A_{32}, A_{28}, A_{30}, A_{33}, A_{25}$  (抢占  $A_{27}$ ),  $A_{27}, A_{20}, A_{26}$  (抢占  $A_{30}$ ),  $A_6, A_{22}, A_{21}, A_{30}, A_{23}, A_{14}, A_{24}, A_{16}, A_{15}, A_{13}, A_{19}$  (抢占  $A_{14}$ ),  $A_{14}, A_{18}, A_{31}, A_{17}$  (抢占  $A_{18}$ ),  $A_{18}, A_7, A_{12}, A_9, A_8, A_{11}, A_4, A_{10}, A_5, A_3, A_2, A_1$ , 共用时为 23. 其调度效果之所以不理想, 是因为该算法只关注工序加工设备的闲忙和工序的路径长度, 没有综合考虑工序的加工用时以及其在工艺树中的相对位置等因素对调度结果的影响. 例如, 工序  $A_{33}$  优先工序  $A_{20}$  调度, 使得工序  $A_{20}$  所在工序序列中

后续工序  $A_{22}, A_{19}$  等与其并行工序的并行性降低, 并影响其他后续工序的调度, 最终影响调度结果.

本文结合工艺树自身特点, 综合考虑工序的加工设备、加工用时及其在工艺树中的相对位置等因素对调度结果的影响. 从不同工序序列上的并行工序的若干组合中, 选择最适合当前调度目标的方案, 事实上扩大了问题的解空间, 较之前算法考虑的更加全面, 在调度产品  $A$  时, 共用时为 20, 结果优于前两种算法.

下面介绍本文算法调度产品  $A$  的调度过程 (详见表 2~5).

表 2 本文算法调度产品  $A$  过程  
Table 2 Scheduling the product  $A$  by the algorithm proposed

工序号	加工设备	准调度时间点	试调度方案加工总时间	后续工序 静态调度时间	确定调度时间点	当前方案中每个工序调度时间点
$A_1$	$M_3$	-	-	-	0	$A_1: 0;$
$A_2$	$M_4$	-	-	-	2	$A_1: 0, A_2: 2;$
$A_{10}$	$M_3$	-	-	-	4	$A_1: 0, A_2: 2, A_{10}: 4;$
$A_{11}$	$M_2$	-	-	-	6	$A_1: 0, A_2: 2, A_{10}: 4, A_{11}: 6;$
$A_{18}$	$M_1$	-	-	-	7	$A_1: 0, A_2: 2, A_{10}: 4, A_{11}: 6,$ $A_{18}: 7;$
$A_{15}$	$M_3$	-	-	-	8	$A_1: 0, A_2: 2, A_{10}: 4, A_{11}: 6,$ $A_{18}: 7, A_{15}: 8;$
$A_{21}$	$M_2$	-	-	-	11	$A_1: 0, A_2: 2, A_{10}: 4, A_{11}: 6,$ $A_{18}: 7, A_{15}: 8, A_{21}: 11;$
$A_{26}$	$M_3$	-	-	-	14	$A_1: 0, A_2: 2, A_{10}: 4, A_{11}: 6,$ $A_{18}: 7, A_{15}: 8, A_{21}: 11, A_{26}: 14;$
$A_{25}$	$M_2$	-	-	-	16	$A_1: 0, A_2: 2, A_{10}: 4, A_{11}: 6,$ $A_{18}: 7, A_{15}: 8, A_{21}: 11, A_{26}: 14,$ $A_{25}: 16;$
$A_{28}$	$M_1$	-	-	-	17	$A_1: 0, A_2: 2, A_{10}: 4, A_{11}: 6,$ $A_{18}: 7, A_{15}: 8, A_{21}: 11, A_{26}: 14,$ $A_{25}: 16, A_{28}: 17;$
$A_3$	$M_3$	2 6 11 16	19 19 18 19	18 22 27 32	2	$A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 7,$ $A_{18}: 8, A_{15}: 9, A_{21}: 12, A_{26}: 15,$ $A_{25}: 17, A_{28}: 18, A_3: 2;$
$A_9$	$M_4$	5 19	19 22	19 32	5	$A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 7,$ $A_{18}: 8, A_{15}: 9, A_{21}: 12, A_{26}: 15,$ $A_{25}: 17, A_{28}: 18, A_3: 2, A_9: 5;$
$A_{13}$	$M_3$	8 12 17	20 19 19	18 22 27	8	$A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 7,$ $A_{18}: 8, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8;$
$A_{16}$	$M_2$	10 16 19	20 21 22	18 24 27	10	$A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 7,$ $A_{18}: 8, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10;$

表 3 本文算法调度产品  $A$  过程 (续表 2)Table 3 Scheduling the product  $A$  by the algorithm proposed (continued Table 2)

工序号	加工设备	准调度时间点	试调度方案加工总时间	后续工序 静态调度时间	确定调度时间点	当前方案中每个工序调度时间点
$A_{23}$	$M_4$	13 20	20 22	20 23	13	$A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 7,$ $A_{18}: 8, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 13;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 7,$
$A_{29}$	$M_1$	15	20	18	15	$A_{18}: 8, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 13, A_{29}: 15;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 7,$
$A_5$	$M_4$	4 8  15 20	20 20  20 21	16 20  27 32	4	$A_{18}: 8, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 13, A_{29}: 15,$ $A_5: 4;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$
$A_7$	$M_2$	5 8  13 16 19	20 21  23 21 22	16 19  24 28 30	5	$A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 13, A_{29}: 15,$ $A_5: 4, A_7: 5;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$
$A_{17}$	$M_1$	8 10 18	20 20 20	16 18 26	8	$A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 13, A_{29}: 15,$ $A_5: 4, A_7: 5, A_{17}: 8;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$
$A_{19}$	$M_4$	9 15 20	20 20 22	16 22 27	9	$A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 13, A_{29}: 15,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 9;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$
$A_{22}$	$M_1$	11 18	20 21	16 23	11	$A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 13, A_{29}: 15,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 9,$ $A_{22}: 11;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$
$A_{20}$	$M_4$	14 15 20	21 20 22	16 17 22	15	$A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 13, A_{29}: 15,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 9,$ $A_{22}: 11, A_{20}: 15;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$
$A_4$	$M_2$	2 8 9  13 16 19	20 23 22  23 21 22	13 19 20  24 28 30	2	$A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 13, A_{29}: 15,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 9,$ $A_{22}: 11, A_{20}: 15, A_4: 2;$

表 4 本文算法调度产品 A 过程 (续表 3)  
Table 4 Scheduling the product A by the algorithm proposed (continued Table 3)

工序号	加工设备	准调度时间点	试调度方案加工总时间	后续工序 静态调度时间	确定调度时间点	当前方案中每个工序调度时间点
$A_8$	$M_1$	5 9 10  14 18	20 22 20 20  20	13 17 18  22  26	5	$A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 13, A_{29}: 15,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 9,$ $A_{22}: 11, A_{20}: 15, A_4: 2, A_8: 5;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 5,$ $A_{14}: 8;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 5,$ $A_{14}: 8;$
$A_{14}$	$M_4$	7 8 11  15 17 20	26 20 20  22 22 24	13 14 17  21  23 26	8	$A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 5,$ $A_{14}: 8;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 5,$ $A_{14}: 8, A_{27}: 16;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 5,$ $A_{14}: 8, A_{27}: 16, A_{12}: 5;$
$A_{27}$	$M_2$	12 13  16 19	21 22  20 21	14 15  18 21	16	$A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 5,$ $A_{14}: 8, A_{27}: 16;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 6,$ $A_{14}: 8, A_{27}: 16, A_{12}: 5;$
$A_{12}$	$M_1$	5 7  9 10  17 20	20 20  21 20  21 21	13 15  17 18  25 28	5	$A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 6,$ $A_{14}: 8, A_{27}: 16, A_{12}: 5;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 6,$ $A_{14}: 8, A_{27}: 16, A_{12}: 5, A_{31}: 9;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16, A_{25}: 18,$ $A_{28}: 19, A_3: 2, A_9: 5, A_{13}: 8,$ $A_{16}: 10, A_{23}: 14, A_{29}: 17, A_5: 4,$ $A_7: 5, A_{17}: 8, A_{19}: 12, A_{22}: 14,$ $A_{20}: 17, A_4: 2, A_8: 6, A_{14}: 8, A_{27}: 16,$ $A_{12}: 5, A_{31}: 9, A_{24}: 10;$
$A_{31}$	$M_2$	6 8  9 13  16 18 19	22 21  20 21  21 21 20	13 15  16 20  23 25  26	9	$A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 6,$ $A_{14}: 8, A_{27}: 16, A_{12}: 5, A_{31}: 9;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16, A_{25}: 18,$ $A_{28}: 19, A_3: 2, A_9: 5, A_{13}: 8,$ $A_{16}: 10, A_{23}: 14, A_{29}: 17, A_5: 4,$ $A_7: 5, A_{17}: 8, A_{19}: 12, A_{22}: 14,$ $A_{20}: 17, A_4: 2, A_8: 6, A_{14}: 8, A_{27}: 16,$ $A_{12}: 5, A_{31}: 9, A_{24}: 10;$
$A_{24}$	$M_1$	10 17  20	20 24  24	16 23  26	10	$A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 6,$ $A_{14}: 8, A_{27}: 16, A_{12}: 5, A_{31}: 9;$ $A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16, A_{25}: 18,$ $A_{28}: 19, A_3: 2, A_9: 5, A_{13}: 8,$ $A_{16}: 10, A_{23}: 14, A_{29}: 17, A_5: 4,$ $A_7: 5, A_{17}: 8, A_{19}: 12, A_{22}: 14,$ $A_{20}: 17, A_4: 2, A_8: 6, A_{14}: 8, A_{27}: 16,$ $A_{12}: 5, A_{31}: 9, A_{24}: 10;$

表 5 本文算法调度产品 A 过程 (续表 4)  
Table 5 Scheduling the product A by the algorithm proposed (continued Table 4)

工序号	加工设备	准调度时间点	试调度方案加工总时间	后续工序 静态调度时间	确定调度时间点	当前方案中每个工序调度时间点
$A_{30}$	$M_3$	14 18	20 20	16 20	14	$A_1: 0, A_2: 2, A_{10}: 5, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 6,$ $A_{14}: 8, A_{27}: 16, A_{12}: 5, A_{31}: 9,$ $A_{24}: 10, A_{30}: 14;$ $A_1: 0, A_2: 2, A_{10}: 6, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 6,$ $A_{14}: 8, A_{27}: 16, A_{12}: 5, A_{31}: 9,$ $A_{24}: 10, A_{30}: 14, A_6: 5;$ $A_1: 0, A_2: 2, A_{10}: 6, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 6,$ $A_{14}: 8, A_{27}: 16, A_{12}: 5, A_{31}: 9,$ $A_{24}: 10, A_{30}: 14, A_6: 5,$ $A_{32}: 13;$ $A_1: 0, A_2: 2, A_{10}: 6, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 6,$ $A_{14}: 8, A_{27}: 16, A_{12}: 5, A_{31}: 9,$ $A_{24}: 10, A_{30}: 14, A_6: 5,$ $A_{32}: 13,$ $A_1: 0, A_2: 2, A_{10}: 6, A_{11}: 8,$ $A_{18}: 9, A_{15}: 10, A_{21}: 13, A_{26}: 16,$ $A_{25}: 18, A_{28}: 19, A_3: 2, A_9: 5,$ $A_{13}: 8, A_{16}: 10, A_{23}: 14, A_{29}: 17,$ $A_5: 4, A_7: 5, A_{17}: 8, A_{19}: 12,$ $A_{22}: 14, A_{20}: 17, A_4: 2, A_8: 6,$ $A_{14}: 8, A_{27}: 16, A_{12}: 5, A_{31}: 9,$ $A_{24}: 10, A_{30}: 14, A_6: 5, A_{32}: 13,$ $A_{33}: 16;$
$A_6$	$M_3$	5 7  10 13  16 18	20 20  21 20  21 20	6 8  11 14  17 19	5	
$A_{32}$	$M_3$	13 16 18	20 21 20	14 17 19	13	
$A_{33}$	$M_4$	16 19  20	20 21  21	17 20  21	16	

步骤 1. 将产品 A 中工序加工偏序关系逆置, 如图 12 所示.

步骤 2. 工序序列排序策略为产品工序确定调度顺序. 第一轮计算产品 A 中各叶子节点的路径长度分别为:  $A_6: 6, A_{20}: 16, A_{28}: 18, A_{32}: 12, A_{27}: 13, A_{30}: 13, A_{33}: 1$  和  $A_{29}: 18$ . 路径最长节点为

$A_{28}$  和  $A_{29}$ , 其中  $A_{28}$  路径上的节点数多于  $A_{29}$ , 选择工序  $A_{28}$  及其路径上全部节点作为第 1 工序序列节点, 按加工偏序关系依次入工序队列, 此时工序队列中工序为:  $A_1, A_2, A_{10}, A_{11}, A_{18}, A_{15}, A_{21}, A_{26}, A_{25}, A_{28}$ , 同时将第 1 工序序列上工序从工艺树中删除, 此时工艺树变成森林; 第二轮计算当前森林

中各叶子节点的路径长度分别为:  $A_6$ : 2,  $A_{20}$ : 12,  $A_{32}$ : 1,  $A_{27}$ : 11,  $A_{30}$ : 11,  $A_{33}$ : 14 和  $A_{29}$ : 16. 具有唯一路径最长节点为  $A_{29}$ . 选择工序  $A_{29}$  及其路径上全部节点作为第 2 工序序列节点, 按加工偏序关系依次入工序队列, 此时工序队列中工序为:  $A_1, A_2, A_{10}, A_{11}, A_{18}, A_{15}, A_{21}, A_{26}, A_{25}, A_{28}, A_3, A_9, A_{13}, A_{16}, A_{23}, A_{29}$ . 依次类推, 直至森林为空. 最终工序队列中工序调度顺序为:  $A_1, A_2, A_{10}, A_{11}, A_{18}, A_{15}, A_{21}, A_{26}, A_{25}, A_{28}, A_3, A_9, A_{13}, A_{16}, A_{23}, A_{29}, A_5, A_7, A_{17}, A_{19}, A_{22}, A_{20}, A_4, A_8, A_{14}, A_{27}, A_{12}, A_{31}, A_{24}, A_{30}, A_6, A_{32}, A_{33}$ .

**步骤 3.** 将最长序列 (第 1 工序序列) 上工序出队形成初始调度方案, 详见表 2.

**步骤 4.** 逐一调度非最长工序序列上工序, 逐步形成产品调度方案, 详见表 3~5.

## 6 结论

考虑后续工序的择时综合调度算法通过提出的 3 个策略达到了优化综合调度的目的, 主要结论如下:

1) 工序序列排序策略从工艺树的整体结构出发安排工序的调度顺序, 提高了串行工序的紧密度;

2) 择时调度策略扩大了问题的解空间, 调度时结合工艺树自身特点灵活选择工序加工开始时间, 兼顾了并行工序的并行性和串行工序的紧密度;

3) 考虑后续工序策略在择时调度策略基础上补充考虑未调度工序, 进一步提高了择时调度策略的精确度;

4) 算法时间复杂度不超过三次多项式.

综上所述, 本文算法为解决一般综合调度问题提供了新的启示, 并为进一步的深入研究拓展了思路、奠定了基础, 具有一定的理论和实际意义.

## References

- Wang Da-Zhi, Liu Shi-Xin, Guo Xi-Wang. A multi-agent evolutionary algorithm for solving total tardiness permutation flow-shop scheduling problem. *Acta Automatica Sinica*, 2014, **40**(3): 548–555  
(王大志, 刘士新, 郭希旺. 求解总拖期时间最小化流水车间调度问题的多智能体进化算法. 自动化学报, 2014, **40**(3): 548–555)
- Huang Min, Fu Ya-Ping, Wang Hong-Feng, Zhu Bing-Hu, Wang Xing-Wei. Job-shop scheduling model and algorithm with machine deterioration. *Acta Automatica Sinica*, 2015, **41**(3): 551–558  
(黄敏, 付亚平, 王洪峰, 朱兵虎, 王兴伟. 设备带有恶化特性的作业车间调度模型与算法. 自动化学报, 2015, **41**(3): 551–558)
- Wang Sheng-Yao, Wang Ling, Xu Ye, Zhou Gang. An estimation of distribution algorithm for solving hybrid flow-shop scheduling problem. *Acta Automatica Sinica*, 2012, **38**(3): 437–443  
(王圣尧, 王凌, 许烨, 周刚. 求解混合流水车间调度问题的分布估计算法. 自动化学报, 2012, **38**(3): 437–443)
- Jia Wen-You, Jiang Zhi-Bin, Li You. Family-oriented to optimize scheduling problem of re-entrant batch processing machine with due window. *Journal of Mechanical Engineering*, 2015, **51**(12): 192–201  
(贾文友, 江志斌, 李友. 面向产品族优化时间窗下可重入批处理机调度. 机械工程学报, 2015, **51**(12): 192–201)
- Zhang Jie, Zhang Peng, Liu Guo-Bao. Two-stage ant colony algorithm based job shop scheduling with unrelated parallel machines. *Journal of Mechanical Engineering*, 2013, **49**(6): 136–144  
(张洁, 张朋, 刘国宝. 基于两阶段蚁群算法的带非等效并行机的作业车间调度. 机械工程学报, 2013, **49**(6): 136–144)
- Qiang Lei, Xiao Tian-Yuan. Model extended BOA to solve hybrid assembly scheduling problems. *Computer Integrated Manufacturing Systems*, 2007, **13**(2): 317–322  
(羌磊, 肖天元. 应用扩展贝叶斯进化算法求解混流装配调度问题. 计算机集成制造系统, 2007, **13**(2): 317–322)
- Wang Hao-Xiang, Yan Hong-Sen, Wang Zheng. Adaptive assembly scheduling of aero-engine based on double-layer Q-learning in knowledgeable manufacturing. *Computer Integrated Manufacturing Systems*, 2014, **20**(12): 3000–3010  
(汪浩祥, 严洪森, 汪峥. 知识化制造环境中基于双层Q学习的航空发动机自适应装配调度. 计算机集成制造系统, 2014, **20**(12): 3000–3010)
- Xie Zhi-Qiang, Liu Sheng-Hui, Qiao Pei-Li. Dynamic job-shop scheduling algorithm based on ACPM and BFSM. *Journal of Computer Research and Development*, 2003, **40**(7): 977–983  
(谢志强, 刘胜辉, 乔佩利. 基于 ACPM 和 BFSM 的动态 Job-Shop 调度算法. 计算机研究与发展, 2003, **40**(7): 977–983)
- Xie Zhi-Qiang, Yang Jing, Yang Guang, Tan Guang-Yu. Dynamic job-shop scheduling algorithm with dynamic set of operation having priority. *Chinese Journal of Computers*, 2008, **31**(3): 502–508  
(谢志强, 杨静, 杨光, 谭光宇. 可动态生成具有优先级工序集的动态 Job-Shop 调度算法. 计算机学报, 2008, **31**(3): 502–508)
- Xie Zhi-Qiang, Yang Jing, Zhou Yong, Zhang Da-Li, Tan Guang-Yu. Dynamic critical paths multi-product manufacturing scheduling algorithm based on operation set. *Chinese Journal of Computers*, 2011, **34**(2): 406–412  
(谢志强, 杨静, 周勇, 张大力, 谭光宇. 基于工序集的动态关键路径多产品制造调度算法. 计算机学报, 2011, **34**(2): 406–412)
- Xie Zhi-Qiang, Xin Yu, Yang Jing. Integrated scheduling algorithm based on event driven by machines' idle. *Journal of Mechanical Engineering*, 2011, **47**(11): 139–147  
(谢志强, 辛宇, 杨静. 基于设备空闲事件驱动的综合调度算法. 机械工程学报, 2011, **47**(11): 139–147)
- Xie Zhi-Qiang, Xin Yu, Yang Jing. Machine-driven integrated scheduling algorithm with rollback-preemptive. *Acta Automatica Sinica*, 2011, **37**(11): 1332–1343  
(谢志强, 辛宇, 杨静. 可回退抢占的设备驱动综合调度算法. 自动化学报, 2011, **37**(11): 1332–1343)
- Xie Z Q, Hao S Z, Ye G J, Tan G Y. A new algorithm for complex product flexible scheduling with constraint between jobs. *Computers and Industrial Engineering*, 2009, **57**(3): 766–772

- 14 Xie Z Q, Yang J, He Y J, Li Z M. An algorithm of simple multi-product scheduling problem with no-wait constraint between operations. *Advanced Materials Research*, 2010, **129–131**: 902–907
- 15 Xie Z Q, Gui Z Y, Yang J. Integrated scheduling algorithm based on dynamic essential short path by device driver. *Journal of Information and Computer Science*, 2013, **10**(4): 1075–1084
- 16 Xie Z Q, Yang J, He Y J, Ye G J. Dynamic integrated scheduling algorithm of complex multi-products with identical machines. *Advanced Materials Research*, 2010, **129–131**: 897–901
- 17 Xie Z Q, He Y J, Liu C H, Yang J. Study on data storage of dynamic integrated scheduling. *Procedia Engineering*, 2012, **29**: 4017–4024
- 18 Xie Z Q, Wang P, Gui Z Y, Yang J. Integrated scheduling algorithm based on dynamic essential short path. *Advances in Intelligent and Soft Computing*, 2012, **169**: 709–715

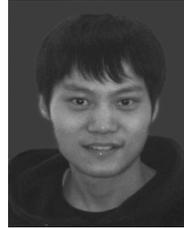


谢志强 博士, 教授, CCF 高级会员. 主要研究方向为企业智能计算与调度系统, 数据处理, 网络优化. 本文通信作者.  
E-mail: xiezhiqiang@hrbust.edu.cn  
(XIE Zhi-Qiang Ph. D., professor, and senior member of China Computer Federation. His research interest covers intelligent computing and scheduling system, data processing and network optimization. Corresponding author of this paper.)

ing system, data processing and network optimization. Corresponding author of this paper.)



张晓欢 哈尔滨理工大学博士研究生. 主要研究方向为企业智能计算.  
E-mail: huanhuan291@126.com  
(ZHANG Xiao-Huan Ph. D. candidate at Harbin University of Science and Technology. Her main research interest is enterprise intelligence computing.)



辛宇 哈尔滨理工大学计算机科学与技术学院讲师. 2015 年获哈尔滨工程大学计算机应用技术博士学位. 主要研究方向为数据库与知识工程.

E-mail: xinyu@hrbeu.edu.cn

(XIN Yu Lecturer at the College of Computer Science and Technology, Harbin University of Science and Technology. He received his Ph. D. degree from Harbin Engineering University in 2015. His research interest covers database and knowledge engineering.)



杨静 博士, 教授, CCF 高级会员. 主要研究方向为数据与知识工程, 数据挖掘, 隐私保护, 软件理论.

E-mail: yangjing@hrbeu.edu.cn

(YANG Jing Ph. D., professor, and senior member of China Computer Federation. Her research interest covers data and knowledge engineering, data mining, privacy protection, and computer software and theory.)