

文章编号: 2095-2163(2022)12-0208-06

中图分类号: TP393

文献标志码: A

基于动态调整叶节点工序的综合调度算法

刘显超¹, 赵男², 杨苗³, 苏艳会⁴, 周伟¹

(1 吉林师范大学 计算机学院, 吉林 四平 136000; 2 四平市第二十五中学, 吉林 四平 136000;

3 吉林师范大学附属学校, 吉林 四平 136000; 4 长春市第十六中学, 长春 130000)

摘要: 针对目前综合调度算法中因忽略叶节点工序的调度优势而导致设备利用率较低、工序间衔接度较弱的问题, 提出了基于动态调整叶节点工序的综合调度算法。算法将复杂产品工艺树中的叶节点工序作为影响综合调度整体优化的关键因素, 通过循环更新和调度叶节点工序达到优化复杂产品综合调度时间成本的目的。

关键词: 综合调度; 叶节点工序; 动态调整

An integrated scheduling algorithm based on dynamically adjusting leaf node processes

LIU Xianchao¹, ZHAO Nan², YANG Zhuo³, SU Yanhui⁴, ZHOU Wei¹

(1 College of Computer Science, Jilin Normal University, Siping Jilin 136000, China; 2 Siping No.25 Middle School, Siping Jilin 136000, China; 3 Affiliated School, Jilin Normal University, Siping Jilin 136000, China; 4 Changchun No.16 Middle School, Changchun 130000, China)

[Abstract] In the existing integrated scheduling algorithms, the convergence degree between processes is weakened and the utilization rate of equipments is reduced due to the neglect of the scheduling advantages of leaf node processes. To solve this problem, an integrated scheduling algorithm based on dynamically adjusting leaf node processes is proposed. The algorithm takes the leaf node processes in the process tree of complex products as an important factor of integrated scheduling and reduces the time cost of integrated scheduling through cyclic updating and scheduling leaf node processes.

[Key words] integrated scheduling; leaf node processes; dynamic adjustment

0 引言

产品的生产制造作为影响企业生产效率的主要因素, 一直都是相关学科研究的热点。在人力资源、设备资源等有限的前提下, 如何更好地提高生产加工的时间效率和设备的利用效率直接决定了企业的生产效率, 也是制造业致力解决的重要问题。

随着互联网技术和计算机技术的发展, 制造行业的产品需求和社会需求也发生了变化^[1-5]。但是, 人们对复杂产品的需求日趋个性化和多样化^[6-7], 因而亟需解决多品种、小批量复杂工艺的产品调度问题。为此, 有专家学者提出了将“产品的加工和装配同步处理”的综合调度^[8]并开展了一系列的研究^[9-12], 提出了诸多调度算法, 也拓展出很多

新的研究领域^[12-18]。

虽然现有的综合调度算法已经取得了较好的研究成果, 但是仍然存在横纵双向优化中兼顾性不完善的问题。例如以“工序”为研究对象的算法中, 文献[13]虽然提出了以工序数量总量来确定调度路径和次序的算法, 但是忽略了叶节点工序的重要作用。文献[14]虽然提出了以串行工序数量为排序策略的算法, 但是忽略了设备利用率对整体调度效果的影响等等。

为解决上述问题, 本文以优化综合调度时间成本为目标, 提出了基于动态调整叶节点工序的综合调度算法。算法以复杂产品工艺树的叶节点工序和对应的加工设备作为研究对象, 首先以复杂产品工艺树原始树图构建叶节点工序和对应加工设备的工

基金项目: 教育部科技司赛尔网络下一代互联网技术创新项目(NGII20180408); 吉林省社科项目(2020C048); 吉林师范大学校级项目“三基一新”型计算机专业人才培养模式研究与实践。

作者简介: 刘显超(1988-), 男, 硕士研究生; 主要研究方向: 网络安全; 赵男(1992-), 女, 硕士, 中学二级教师, 主要研究方向: 翻译英语; 杨苗(1982-), 女, 学士, 小学高级教师, 主要研究方向: 数学建模; 苏艳会(1970-), 女, 学士, 中学一级教师, 主要研究方向: 建模分析; 周伟(1979-), 女, 博士研究生, 吉林省高校重点实验室办公室主任, 助理研究员, 主要研究方向: 计算机应用技术。

通讯作者: 周伟 Email: 867458539@qq.com

收稿日期: 2022-03-27

序组,在工序组中优先调度层优先级较高的叶节点工序;其次,在工艺树中删除已调度的叶节点工序,重构复杂产品工艺树和叶节点工序,更新叶节点工序和对应加工设备的工序组;如此循环分解,直到原始工艺树中所有工序全部调度完毕的综合调度方法。

1 相关定义

假设综合调度系统有 m 台加工设备,需要加工完成复杂产品的 n 道工序,要求如下:

- (1) 在加工的某个时刻,加工设备只能连续加工对应的工序。
- (2) 反之,一道工序也必须在对应的设备上加工。
- (3) 不存在相同设备。

定义 1 叶节点工序 在复杂产品工艺树中,将没有紧前工序约束、但是具有紧后约束工序的工序定义为叶节点工序,且叶节点工序数量必须大于等于 1。

定义 2 工序层优先级 根据文献[20]首次提出的层优先级问题,描述为:对于具有 n 层结构的复杂产品,首先将根节点工序的优先级设为最低,定义为 1,根节点工序的所有后裔节点工序的优先级定义为 2,以此类推,直到第 n 层的所有节点的优先级设为最高,定义为 n [19]。

定义 3 工序组 在综合调度中,处于当前复杂产品工艺树中的叶节点工序和其对应的加工设备组成的加工组合定义为工序组。在工序组中,一台设备至少对应一道叶节点工序。

2 算法设计

复杂产品加工的全过程中,所有加工设备彼此独立,设备间不存在优先顺序的问题。本文分别从横纵两个方面优化了复杂产品的综合调度效果,其中纵向优化是通过在工序组调度的叶节点工序实现的,达到缩短整体加工用时的优化目标;横向优化是通过循环产生新叶节点工序实现的,即通过提高串行工序的紧密衔接度,减少对应设备上工序加工的间隔时间,达到提高各个设备利用率的优化效果。算法流程如图 1 所示,具体阐述如下:

- Step 1** 以复杂产品工艺树原始树图为基础,查找所有叶节点工序。
- Step 2** 建立叶节点工序组。
- Step 3** 判断工序组中待调度叶节点工序是否唯一,是则调度;否则转 Step4。
- Step 4** 按照层优先级由高到低的顺序依次调

度组内叶节点工序。

Step 5 在复杂产品工艺树原始树图中删除已经调度完成的叶节点工序,形成新的工艺树。

Step 6 重复 Step1,直到根节点工序调度完毕。

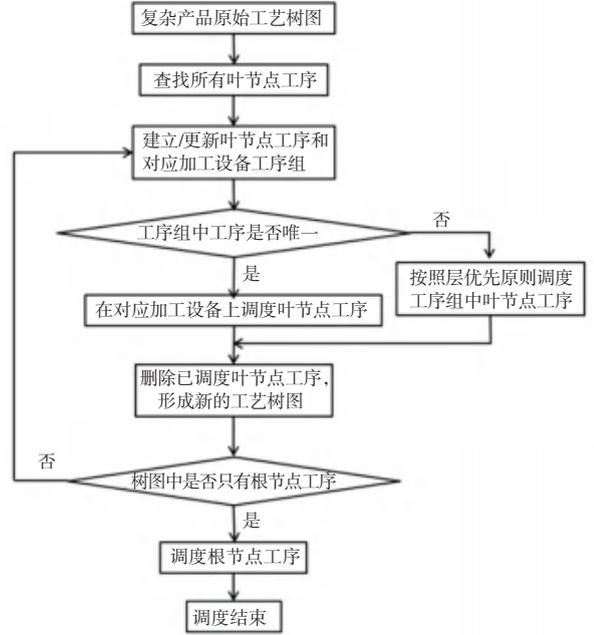


图 1 算法流程图

Fig. 1 Flow chart of the algorithm

3 调度实例演示

现假设复杂产品 A 加工工艺树如图 2 所示。需要说明的是,此算例不特指某一类产品,对于其他小批量、多品种的复杂产品亦适用。对实例流程拟做阐释分述如下。

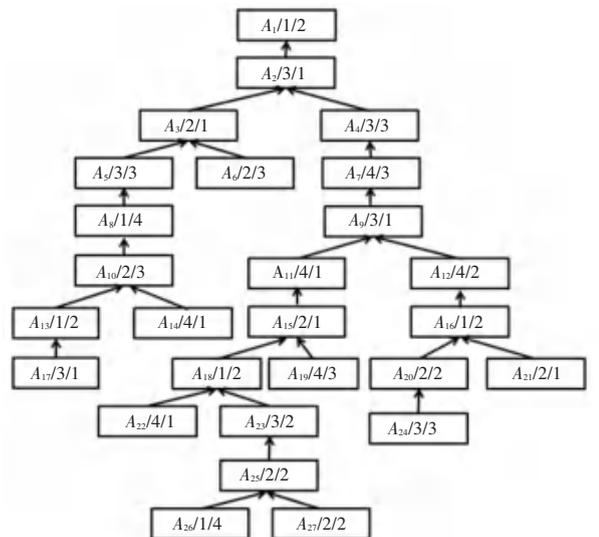


图 2 复杂产品 A 工艺树图

Fig. 2 Complex product A process tree

Step 1 以复杂产品工艺树原始树图为基础,确定原始工艺树图叶节点工序,共计9道,如图3所示。

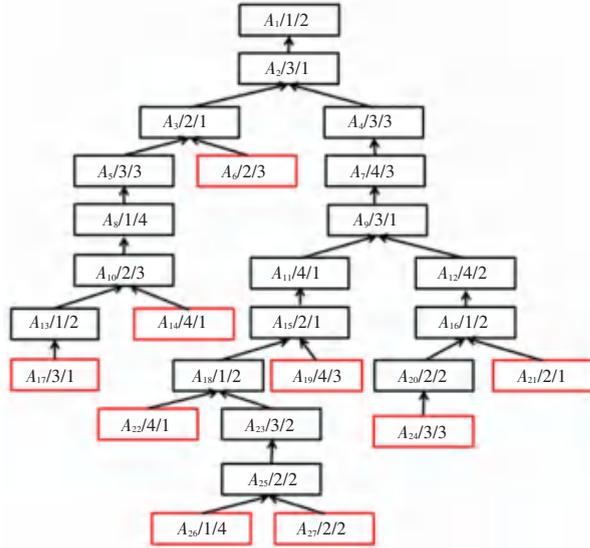


图3 复杂产品A工艺树图初始叶节点工序图

Fig. 3 Initial leaf node process diagram of complex product A process tree diagram

Step 2 建立叶节点工序和对应加工设备工序组,可表示为: $M_1: \{A_{26}/1/4\}; M_2: \{A_6/2/3, A_{21}/2/1, A_{27}/2/2\}; M_3: \{A_{17}/3/1, A_{24}/3/3\}; M_4: \{A_{14}/4/1, A_{19}/4/3, A_{22}/4/1\}$ 。

Step 3 设备 M_1 工序组中待调度叶节点工序

只有 A_{26} ,直接调度;设备 M_2, M_3, M_4 对应的工序组中,叶节点工序不唯一,则按照层优先级由高到低的顺序依次调度组内叶节点工序,得到各工序组调度顺序为 $M_1: \{A_{26}/1/4\}; M_2: \{A_{27}/2/2, A_{21}/2/1, A_6/2/3\}; M_3: \{A_{24}/3/3, A_{17}/3/1\}; M_4: \{A_{22}/4/1, A_{19}/4/3, A_{14}/4/1\}$ 。

Step 4 在复杂产品工艺树原始树图中删除已经调度完成的叶节点工序,形成新的工艺树,得到新的叶节点工序3道,如图4所示。

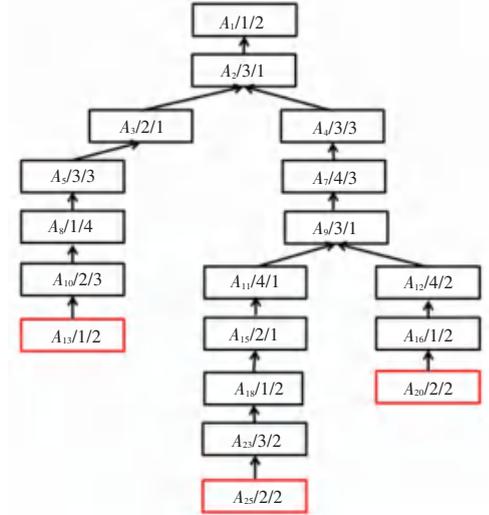


图4 第一次调度结束后的新树图

Fig. 4 New tree after the first scheduling

Step 5 重复上述步骤,直到根节点工序 A_1 调度完毕,调度过程见表1。

表1 复杂产品A调度过程表

Tab. 1 Scheduling process of complex product A

动态调整次数	叶节点工序	工序组	组内工序是否唯一	组内工序调度顺序
1	$A_{26}, A_6, A_{21}, A_{27}, A_{17}, A_{24}, A_{14}, A_{19}, A_{22}$	$M_1: \{A_{26}/1/4\}; M_2: \{A_6/2/3, A_{21}/2/1, A_{27}/2/2\}; M_3: \{A_{17}/3/1, A_{24}/3/3\}; M_4: \{A_{14}/4/1, A_{19}/4/3, A_{22}/4/1\}$	否	$M_1: \{A_{26}/1/4\}; M_2: \{A_{27}/2/2, A_{21}/2/1, A_6/2/3\}; M_3: \{A_{24}/3/3, A_{17}/3/1\}; M_4: \{A_{22}/4/1, A_{19}/4/3, A_{14}/4/1\}$
2	A_{13}, A_{25}, A_{20}	$M_1: \{A_{13}/1/2\}; M_2: \{A_{20}/2/2, A_{25}/2/2\}$	否	$M_1: \{A_{13}/1/2\}; M_2: \{A_{25}/2/2, A_{20}/2/2\}$
3	A_{10}, A_{23}, A_{16}	$M_1: \{A_{16}/1/2\}; M_2: \{A_{10}/2/3\}; M_3: \{A_{23}/3/2\}$	是	$M_1: \{A_{16}/1/2\}; M_2: \{A_{10}/2/3\}; M_3: \{A_{23}/3/2\}$
4	A_8, A_{18}, A_{12}	$M_1: \{A_8/1/4, A_{18}/1/2\}; M_4: \{A_{12}/4/2\}$	否	$M_1: \{A_{18}/1/2, A_8/1/4\}; M_4: \{A_{12}/4/2\}$
5	A_5, A_{15}	$M_2: \{A_{15}/2/1\}; M_3: \{A_5/3/3\}$	是	$M_2: \{A_{15}/2/1\}; M_3: \{A_5/3/3\}$
6	A_3, A_{11}	$M_2: \{A_3/2/1\}; M_4: \{A_{11}/4/1\}$	是	$M_2: \{A_3/2/1\}; M_4: \{A_{11}/4/1\}$
7	A_9	$M_3: \{A_9/3/1\}$	是	$M_3: \{A_9/3/1\}$
8	A_7	$M_4: \{A_7/4/3\}$	是	$M_4: \{A_7/4/3\}$
9	A_4	$M_3: \{A_4/3/3\}$	是	$M_3: \{A_4/3/3\}$
10	A_2	$M_3: \{A_2/3/1\}$	是	$M_3: \{A_2/3/1\}$
11	A_1	$M_1: \{A_1/1/2\}$	是	$M_1: \{A_1/1/2\}$

复杂产品 A 按照动态调整叶节点工序方法, 调度顺序为: $\{A_{26}, A_{27}, A_{21}, A_6, A_{24}, A_{17}, A_{22}, A_{19}, A_{14}, A_{13}, A_{25}, A_{20}, A_{16}, A_{10}, A_{23}, A_{18}, A_8, A_{12}, A_{15}, A_5, A_3, A_{11},$

$A_9, A_7, A_4, A_2, A_1\}$, 调度甘特图如图 5 所示, 共计 27 个加工用时。

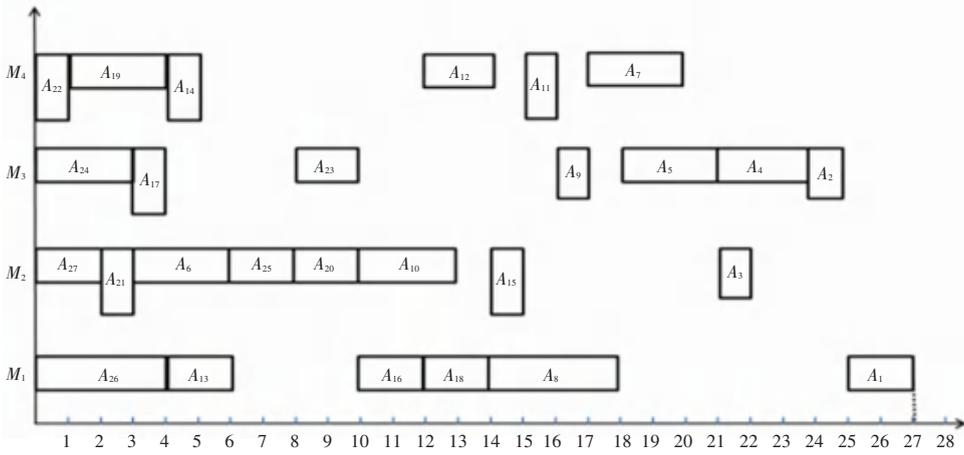


图 5 本文算法加工复杂产品 A 甘特图

Fig. 5 Gantt chart of complex product A processed by this algorithm

4 算法对比分析

4.1 与文献[13]的算法对比

文献[13]的算法, 按照紧密程度将复杂产品的所有工序分为 2 类, 对于紧密度较高的工序组采用

“首次适应调度”算法, 对于紧密度较弱的工序组使用“拟关键路径法”, 调度顺序为 $\{A_{24}, A_{21}, A_{26}, A_{27}, A_{25}, A_{23}, A_{22}, A_{19}, A_{18}, A_{15}, A_{11}, A_{20}, A_{16}, A_{12}, A_9, A_{17}, A_{14}, A_{13}, A_{10}, A_8, A_5, A_6, A_3, A_7, A_4, A_2, A_1\}$, 如图 6 所示, 总用时 28 工时。

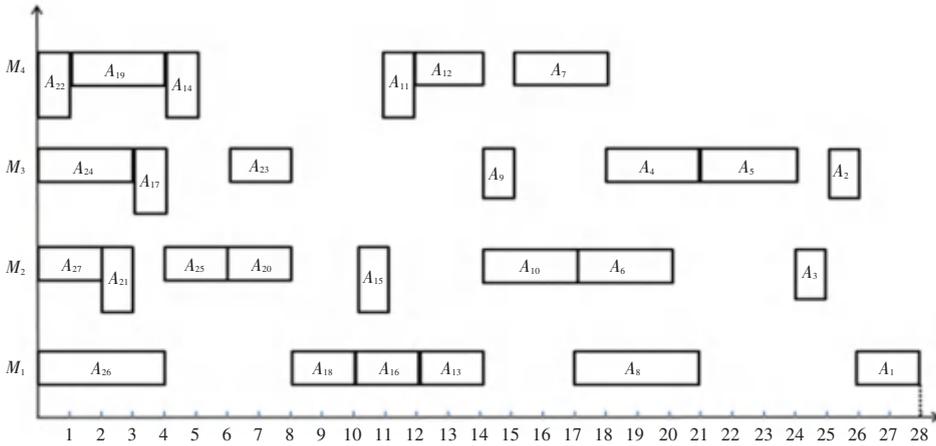


图 6 文献[13]算法甘特图

Fig. 6 Gantt chart of the algorithm in Ref.[13]

对比分析本文算法和文献[13]的算法对应的加工甘特图 5 和图 6 可知, 本文算法中各个加工设备不相同且彼此独立, 所以在满足紧前工序(组)约束前提下, 设备 M_3 在 $t = 16$ 时刻调度工序 A_9 , 充分利用了设备 M_3 在 $t = 16$ 到 $t = 18$ 时刻的空闲段。同时, 因工序 A_5 在图 5 中比在图 6 中早 3 个工时开始加工, 其紧后工序(组) A_3, A_2, A_6 的加工时间比在图 6 中分别提前了 3、1、1 个工时开始加工。在设备 M_2 上, 本文算法从 $t = 0$ 到 $t = 13$ 时刻一直连续加工, 处于“设备忙”的状态, 提高了 M_2 的设备利用率。在

设备 M_1 上本文算法因工序 A_{13} 从 $t = 4$ 时刻开始加工, 比图 6 中的 $t = 12$ 时刻提前了 8 个工时开始加工, 其后续工序 A_8, A_3 分别提前了 3 个工时开始加工。对比分析表明, 本专利方法能够有效减少对对应加工设备的空闲时间, 并提高了加工工序的紧密衔接度, 进而优化了复杂产品整体调度效果。

4.2 与文献[14]算法对比

文献[14]算法在复杂产品工艺树的整体结构的基础上, 首先将工艺树按照排序策略划分为只具有串行关系的工序序列, 然后从调度方案集合中按

照择时策略依次选择加工总用时最小和最早的方案进行调度。

针对复杂产品 A 的加工工艺树,采用文献[14]算法形成初始调度方案为 $\{A_1, A_2, A_4, A_7, A_9, A_{11}, A_{15},$

$A_{18}, A_{18}, A_{25}, A_{26}\}$,在此基础上按照 $\{A_3, A_{11}, A_9, A_{13}, A_{17}, A_{19}, A_{16}, A_{20}, A_{24}, A_6, A_{19}, A_{27}, A_{21}, A_{14}, A_{22}, A_{26}\}$ 的顺序进行调整,结果甘特图如图 7 所示,总加工用时为 31 工时。

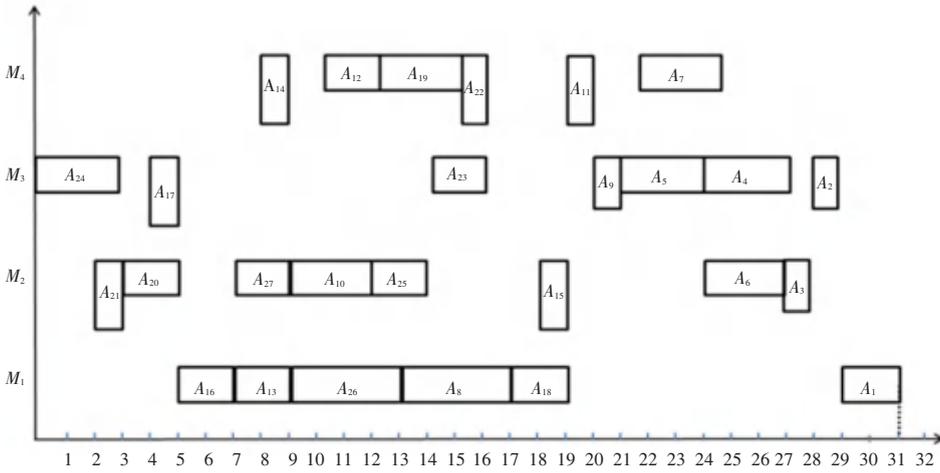


图 7 文献[14]算法甘特图

Fig. 7 Gantt chart of the algorithm in Ref.[14]

对比分析本文算法和文献[14]的算法对应的加工甘特图 5 和图 7 可知,本文算法所有设备均从 $t = 0$ 时刻的始点开始加工,整体上比文献[14]算法中的各工序衔接度要高。

在设备 M_4 上,图 5 在 $t = 20$ 时刻完成了对应所有工序的加工,而图 7 中在设备 M_4 上所有工序加工结束前在 $t = 0$ 至 $t = 8$ 时刻、 $t = 9$ 至 $t = 10$ 时刻、 $t = 16$ 至 $t = 19$ 时刻、 $t = 20$ 至 $t = 22$ 时刻出现共计 14 个工时的设备空闲时间段,明显多于图 5 中的 9 个工时的空闲时间段,设备 M_4 利用率提高了 11%。

在设备 M_3 上,图 5 在 $t = 25$ 时刻完成了对应所有工序的加工,而图 7 中在设备 M_3 上所有工序加工结束前在 $t = 3$ 至 $t = 4$ 时刻、 $t = 5$ 至 $t = 14$ 时刻、 $t = 16$ 至 $t = 20$ 时刻、 $t = 27$ 至 $t = 28$ 时刻出现共计 15 个工时的设备空闲时间段,多于图 5 中的 11 个工时的空闲时间段,设备 M_3 利用率提高了 7.7%。

在设备 M_2 上,图 5 在 $t = 22$ 时刻完成了对应所有工序的加工,而图 7 中在设备 M_2 上所有工序加工结束前在 $t = 0$ 至 $t = 2$ 时刻、 $t = 5$ 至 $t = 7$ 时刻、 $t = 14$ 至 $t = 18$ 时刻、 $t = 19$ 至 $t = 24$ 时刻出现共计 13 个工时的设备空闲时间段,明显多于图 5 中 $t = 13$ 至 $t = 14$ 时刻、 $t = 15$ 至 $t = 21$ 时刻共计 7 个工时的空闲时间段,设备 M_2 利用率提高了 14.6%。

在设备 M_1 上,图 5 在 $t = 27$ 时刻完成了对应所有工序的加工,而图 7 中在设备 M_1 上所有工序加工结束前在 $t = 0$ 至 $t = 5$ 时刻、 $t = 19$ 至 $t = 29$ 时刻出

现共计 15 个工时的设备空闲时间段,多于图 5 中 $t = 6$ 至 $t = 10$ 时刻、 $t = 18$ 至 $t = 25$ 时刻共计 11 个工时的空闲时间段,设备 M_1 利用率提高 7.7%。

综上,从各个设备利用率的角度分析,本文算法采用将对应设备与动态调整产生的叶节点工序进行组合的方法,是从工序和设备两个角度进行优化,对比分析表明本文算法不仅减少了各个设备的空闲时间段,而且提高了所有加工设备的利用率,进而提高了复杂产品加工的设备总体利用率。3 种算法的调度过程时间对比和设备利用率对比见表 2。

表 2 3 种算法的调度过程时间对比和设备利用率对比

Tab. 2 Comparison of scheduling process time and equipments utilization of three algorithms

	空闲时间 总和	产品 A 加工 总工时	设备 利用率/%	设备利用率 相对提高率/%
文献[7]算法	40	28	58.8	0.8
文献[8]算法	56	31	49.1	10.5
本文算法	29	27	59.6	-

4.3 3 种算法调度结果对比分析

对图 1 中的复杂产品 A,文献[13]的算法总加工用时为 28 工时、文献[14]的算法总加工用时为 31 工时,而本文算法总加工用时为 27 工时。由此可知,本文算法更优,主要是因为:

文献[13]的紧密衔接工序组联动的算法虽然注重衔接紧密的工序组,但工序组中只由工序组成,不仅忽略了紧密衔接度不高的其他工序的相对位置

等因素对调度结果的整体影响,而且没有充分利用设备的空闲时间段。例如,设备 M_2 在 $t = 17$ 至 $t = 23$ 时刻、设备 M_3 在 $t = 8$ 至 $t = 14$ 时刻和设备 M_4 在 $t = 5$ 至 $t = 11$ 时刻出现较长时间的空闲。

文献[14]中考虑串行工序紧密度的择时算法与紧密衔接工序组联动的算法相似,也是只以工序为主要优化对象,通过择时调度策略确定串行工序加工开始时间点,同样没有充分考虑到设备的利用率问题,忽略了设备因素在综合调度中的重要作用。例如设备 M_1 在 $t = 18$ 至 $t = 29$ 时刻、设备 M_3 在 $t = 5$ 至 $t = 14$ 时刻、设备 M_4 在 $t = 1$ 至 $t = 8$ 时刻这些时间段一直处于空闲状态,没有加工工序,因此就给复杂产品加工过程造成一定的整体影响。

相对于紧密衔接工序组联动算法、考虑串行工序紧密度的择时算法,本文算法的设备利用率分别提高了 0.8%、10.5%,实验表明本文算法在复杂产品的综合调度中效果更优,不仅为解决一般复杂产品的综合调度提供了一种新的方法,而且为进一步深入研究综合调度拓展了思路,具有一定的理论和实际意义。

5 结束语

综上所述,本文以优化复杂产品综合调度的时间成本目标,以复杂产品工艺树结构中具有重要作用的叶节点工序为优化对象,提出了基于动态调整叶节点工序的算法。实验对比分析表明,本文算法有效地提高了设备的利用率,为综合调度提供了一种新的研究方法。

参考文献

- [1] 程曦,宋铁成. 基于改进多目标布谷鸟搜索的资源调度算法[J]. 计算机应用与软件,2022,39(03):241-246,253.
- [2] 薛建彬,王泽森,张永刚. 边缘计算中基于混沌理论的任务调度优化策略[J]. 华中科技大学学报(自然科学版),2022,50(03):

18-23.

- [3] 冯树民,廖嘉雯,赵璇. 基于乘客换乘需求的重叠线路公交调度优化[J]. 哈尔滨工业大学学报,2022,54(03):41-47.
- [4] 沈彪,沈立炜,李弋. 空间包任务的路径动态调度方法[J]. 计算机科学,2022,49(02):231-240.
- [5] 蒋敬轩,张著洪. 基于免疫粒子群算法的分层注水流量调度[J]. 智能计算机与应用,2021,11(12):72-77.
- [6] 高一龙. 存在柔性加工设备的二车间综合调度算法研究[D]. 哈尔滨:哈尔滨理工大学,2017:13-14.
- [7] 薛静. 协作环境下基于 WBS-FDSM-RAM 的复杂产品项目进度控制研究[D]. 镇江:江苏科技大学,2014.
- [8] 谢志强. 工件间有约束的复杂产品工序调度研究[D]. 哈尔滨:哈尔滨理工大学,2009.
- [9] 朱天浩. 存在多工序同时结束的综合调度算法研究[D]. 哈尔滨:哈尔滨理工大学,2013.
- [10] <https://wenku.baidu.com/view/9af9a4362a160b4e767f5acfa1c7aa00b42a9d70.html>.
- [11] 郭伟飞,雷琦,宋豫川,等. 基于虚拟零部件的零等待约束复杂产品综合调度算法[J]. 机械工程学报,2020,56(04):246-257.
- [12] 张伟涛. 前移存在调整时间综合调度工序的研究[D]. 哈尔滨:哈尔滨理工大学,2012.
- [13] 谢志强,滕宇峥,杨静. 紧密衔接工序组联动的综合调度算法[J]. 自动化学报,2011,37(03):371-379.
- [14] 谢志强,张晓欢,高一龙,等. 考虑串行工序紧密度的择时综合调度算法[J]. 机械工程学报,2018,54(06):191-202.
- [15] 谢志强,周伟,杨静. 考虑层级调度次序的资源协同综合调度算法[J]. 计算机集成制造系统,2022,28(11):3391-3402.
- [16] 谢志强,周伟,余泽睿. 动态调整设备维护开始时间的综合调度算法[J]. 机械工程学报,2021,57(04):240-246.
- [17] XIE Zhiqiang, YANG Dan, MA Mingrui, et al. An improved Artificial Bee Colony Algorithm for the flexible integrated scheduling problem using networked devices collaboration [J]. International Journal of Cooperative Information Systems, 2020,29(3):2040003-2040022.
- [18] GAO Yilong, XIE Zhiqiang, LIU Xinyang, et al. Integrated scheduling algorithm based on the priority constraint table for complex products with tree structure [J]. Advances in Mechanical Engineering, 2020, 12(12):1-15.
- [19] XIE Zhiqiang, YANG Guang, TAN Guangyu. An algorithm of JSSP with dynamic collection of job with priority [C]// Proceedings of the International Technology and Innovation Conference 2006 - Advanced Manufacturing Technologies. Hangzhou, China:UK IEE, 2006:106-111.