DOI: 10.3901/JME.2021.17.217

存在预启动设备的综合调度算法*

谢志强 吕 妮

(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)

摘要: 针对目前综合调度研究中没有考虑存在预启动设备的情况,忽略预启动时间对整体加工时间影响的问题,提出存在预启动设备的综合调度算法。该算法在预启动设备处于空闲状态时采用提出的预启动长路径优先策略,在预启动设备处于工作状态时采用提出的预启动短路径优先策略,以提高预启动设备的并行化和减少了设备的预启动次数;采用提出的调整序列确定策略,确定产生设备空闲时间段的工序为调整目标;采用提出的可调序列工序调整策略,对产生设备空闲的工序的前序工序按层序对其加工时间进行调整,以提高设备的利用率。实例说明提出的算法解决了存在预启动设备的综合调度问题,有一定理论和实际意义。

关键词: 预启动设备; 预启动路径; 预启动时间; 加工时间调整; 综合调度算法

中图分类号: TP278

Integrated Scheduling Algorithm with Pre-start Device

XIE Zhiqiang LÜ Ni

(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080)

Abstract: Since the current pre-start equipment is not considered in the study of comprehensive scheduling, and the effect of pre-start time on the overall processing time is ignored, a comprehensive scheduling algorithm with pre-start equipment is proposed. The algorithm adopts the proposed pre-start long path priority strategy when the pre-start device is in the idle state, and uses the proposed pre-start short path priority strategy when the pre-start device is in the working state to improve the parallelization of the pre-start device and reduce the device The number of pre-launches; use the proposed adjustment sequence determination strategy to determine the process that generates the equipment idle time period as the adjustment target; Adopting the proposed adjustable sequence process adjustment strategy, the processing time of the predecessor process that generates equipment idleness is adjusted according to the sequence to improve the utilization rate of equipment. An example shows that the algorithm proposed solves the problem of comprehensive scheduling of pre-start devices, and has certain theoretical and practical significance.

Key words: pre-start device; pre-start path; pre-start time; processing time adjustment; integrated scheduling algorithm

0 前言

调度是提高企业生产效率的关键,有效的调度优化算法能将企业现有的软硬件资源实现价值最大化和利用率最大化^[1]。车间调度优化问题^[2]最早被 JOHNSON 提出,而且国外的学者针对车间调度问题做出了许多研究^[3-4];人工智能领域研究体系日益成熟,学者们发表了许多利用人工智解决车间调度

优化问题的成果^[5-7];目前,国内的许多学者针对车间调度优化问题也做了大量的研究^[8-12];但这些研究大部分针对的是生产和装配分离的传统车间调度问题。随着科技的进步,对产品多元化的需求促使产品生产趋向多品种小批量,当制造产品品种单一,制造过程是工件间有约束加工、装配,单件复杂产品制造,此时如果仍然将加工任务和装配任务分开单独进行,必然破坏制造过程中加工与装配内在的并行关系,影响产品制造效率^[13],为解决生产与装配统一的调度问题,将装配也转化为加工过程中的一道工序,谢志强等提出了加工与装配同时推进的

* 国家自然科学基金(61772160)资助项目。20200604 收到初稿,20201226 收到修改稿

综合调度^[14]。目前对综合调度的研究主要分为基于工序和基于设备的两类研究。

在基于工序的研究中,文献[14]中的拟关键路径法从纵向角度分析产品加工树中工序间的密切联系,提高关键设备的紧凑性;文献[15]则从横向角度提出了层优先、短用时、长路径的动态调整策略,注重了横向工序之间的并行性;在基于工序的研究中也考虑到一些工序间存在特殊约束的综合调度问题,例如通过逆序调度解决了多工序同时结束问题。在基于设备的研究中,文献[16]针对基于工序研究中由于对工序和加工设备之间的内在联系而产生的大量设备空闲时间的问题,提出了基于设备空闲事件驱动的综合调度算法,文献[17]在文献[16]的基础上采用回退策略优先调度对调度结果影响重要的长路径工序,提高了设备的并行性;在基于设备研究中也解决了一些柔性调度问题,例如文献[18]利用设备驱动解决了柔性调度中的可重叠加工问题。

综上所述,目前没有针对存在预启动设备的综合调度问题进行的研究,事实上在实际加工生产中会存在一些设备需要在加工之前进行一些预热处理,例如精密机床在对工件进行加工之前需要先空转预热达到热平衡条件,将这类过程称之为预启动。预启动设备如果反复多次进行预启动,会造成时间以及设备资源的浪费,因此减少设备的预启动次数是解决存在预启动设备的综合调度问题的一个关键点。为了减少预启动设备的预启动次数以及提高预启动设备的并行性,提出了预启动路径优先设备驱动调度策略。该策略在预启动设备处于空闲状态时采用提出的预启动长路径优先策略,使预启动设备尽可能晚一点进入加工状态,使预启动设备上的工序更加紧凑,以减少预启动设备的预启动次数;当预启动设备处于工作状态时采用提出的预启动短路径优先策略,使预启动设备上的工序在其工作状态时尽早到达预启动设备,以减少预启动次数,并达到提高预启动设备并行性的目的。设备驱动在寻找可调度工序时总是在当前可调度并行工序中寻找,增加了串行工序之间的等待时间,破坏了工序紧密串行联系,为此提出了针对设备驱动结果进行基于工序串行的可调序列工序调整策略,即对按照可调工序确定策略寻找到的可调工序序列中的工序通过关键路径对其开始加工时间进行调整。本文算法将二者相结合,先通过预启动路径优先设备驱动调度策略提高设备的并行性,然后通过可调工序调整策略兼顾了工序间的串行联系。

1 问题描述分析

在研究综合调度问题时,一般将加工工序和装配工序都统一看作是工序,将加工设备和装配设备都统一视为设备。在一般综合调度问题的研究中,要求每道工序有且仅有一台设备可以进行加工;一台设备一旦开始加工某道工序,必须加工完毕该工序才可以开始加工其他的工序;每道工序开始加工之前他的所有紧前工序必须加工完毕或者没有紧前工序才可以可以在其对应的加工设备上加工;每道工序的加工时间确定,且与其被安排加工的先后顺序无关;工序之间可以等待,即两道平行工序需要竞争同一设备时可以等待其中一个完成加工另一个工序再占用该设备;设备在工序到达之前可以空闲,等待之前所有工序完成后再加工需要该设备的工序;最后一道工序加工完成的时间为产品加工总时间。

为了更详细地描述存在预启动设备的综合调度算法,先介绍几个该算法的相关概念。

定义1 部分加工树:将产品加工工艺树中的某节点虚拟为根节点,该节点与其前序节点共同构成的该节点的产品加工工艺树。

定义2 预启动设备:在对工件进行正式加工之前,需要进行预启动操作的设备。

定义3 预启动时间:设备预启动阶段所需要的时间,同一设备的预启动时间固定不变。

定义4 预启动路径长度:从叶子节点到第一个需要在预启动设备上加工的节点的路径长度。

定义5 可调工序序列:在通过可调序列确定策略得到的,可以对其开始加工再次进行调整的工序的集合。

存在预启动设备的综合调度算法主要目的是通过减少预启动时间对整体加工时间的影响来节省时间提高工作效率,也更加节省能源。该算法先按照预启动路径优先设备驱动策略对加工树上的所有工序进行调度,该策略在预启动设备处于空闲状态时采用提出的预启动长路径优先策略,使预启动设备尽可能晚一点进入加工状态,使预启动设备上的工序更加紧凑,以减少预启动设备的预启动次数;当预启动设备处于工作状态时采用提出的预启动短路径优先策略,使预启动设备上的工序在其工作状态时尽早到达预启动设备,以减少预启动次数,并达到提高预启动设备并行性的目的;然后通过可调序

列确定策略确定可以进行调整的工序序列; 最后通过可调序列工序调整策略对可调工序序列中的工序按照关键路径对其开始加工时间进行调整, 实现对预启动路径优先设备驱动调度策略调度结果的优化, 最终得到一个可行有效的调度方案。

产品加工完毕的时间为各设备完工的最大时间值, 因此在给定的约束条件下的最短时间 T 是所求的结果, 设现在需要加工的某产品共有 N 道工序, 其中有 $L(L < N)$ 道工序需要在预启动设备上加工, 且需要在 M 台设备上完成加工, 其数学模型描述为

$$T = \min[\max(t_{mn})] \quad (1)$$

$$\text{s.t.} \min(x_{ij})$$

$$T \geq \lim \sum (i_j + g_{ij} + y_i) \quad (2)$$

$$i=1, 2, \dots, N, j=1, 2, \dots, M$$

式中, t_{mn} 为在第 m 台设备上第 n 道工序的完成时间; x_{ij} 为第 i 台设备上第 j 道工序开始加工的时间; i_j 为第 i 台设备上第 j 道工序连续加工时间; g_{ij} 为第 i 台设备上第 j 道工序开始时间和第 $j-1$ 道工序结束时间的差; y_i 为第 i 台设备在正式开启加工之前所需要的预启动时间。

由上述公式可知, 每道工序的固定加工时间是不可变的, 尽量降低预启动时间和设备空闲时间对总加工时间的影响是解决问题的关键, 为此提出了存在预启动设备的综合调度算法。

2 解决问题策略

一般调度算法研究的设备都是只要设备空闲工序无需等待, 工序到达设备即可开始加工, 而在实际加工过程中存在一些设备需要进行预启动操作, 达到设定的标准条件之后才可以对工序进行加工。本文算法针对预启动设备在加工调度过程中对工序和设备间的调度进行了研究和分析, 从而找到一种既节省时间又减少预启动次数的调度算法。

2.1 预启动长路径优先策略

从某节点到第一个需要在预启动设备上加工的工序的路径长度为该节点的预启动路径长度。预启动长路径优先策略的思想如下所述:

当设备驱动时刻, 存在两个及两个以上工序可供设备选择时, 优先选择预启动路径长的工序在设备上优先加工, 这样增加了预启设备的并行性; 当

预启动路径长度相同时可优先调度加工时间长的工序, 可以避免预启动设备过早进入空闲状态, 因为预启动设备一旦不能连续加工, 在下一个工序进行加工之前则需要对预启动设备重新进行预启动, 所以这样可以避免预启动设备的反复多次预启动造成不必要的资源以及成本的浪费。

2.2 预启动路径优先策略

在预启动设备处于非空闲状态时, 为了使需要在预启动设备上加工工序尽早到达预启动设备, 采用预启动短路径优先策略, 策略的核心思想如下:

当多个工序竞争同一个设备时, 优先调度到预启动设备路径短的工序, 这样可以减少预启动设备空闲状态的时间, 当预启动路径相同时则优先调度加工时间短的工序, 这样可以使设备更早得到释放, 使设备状态尽早转化, 可以加工其他工序。

2.3 可调序列确定策略

在利用预启动路径优先的设备驱动策略对所有工序进行排列时, 可能会造成其他设备长时间处于空闲状态, 降低了设备的利用率。利用可调序列确定策略可以确定最先出现空闲时间段的设备, 并且确定可以进行调整的序列。

首次使用可调序列确定策略的初始序列是由预启动路径优先的设备驱动策略所得到的, 之后每次调用可调序列确定策略的工序序列都是通过可调序列工序调整策略调整之后的调度序列。首先通过设备驱动时刻以及设备的状态确认最先出现空闲时间段的设备, 首次出现空闲时间段的设备, 在该时间段之后加工的工序可被视为可调序列, 每个设备在完成可调序列调整之后, 在下次调用可调序列确认策略时, 该设备上的空闲时间需要被忽略。通过该策略可以确定针对设备空闲时间段每次需要进行调整的工序序列。

2.4 可调序列工序调整策略

可调序列工序调整策略针对由可调序列确定策略确定的可调序列中的工序进行调整。首先要将可调序列中的工序虚拟为根节点形成以其为根的部分加工树, 然后计算每棵部分加工树的关键路径, 并按照关键路径升序排序, 优先调整关键路径短的部分加工树上的工序, 选定调整的部分加工树上的工序按照层级优先进行调整, 工序前移调整插入的设备驱动时刻不能影响移动工序的加工设备上其他工序的后序工序的开始加工时刻。

工序首个可插入时刻需要满足两个条件, 一是

该时刻必须是该工序紧前工序完成时候的某个设备驱动时刻,二是该工序插入该时刻之后不能影响该设备上其他工序的后序工序的开始时刻。如果不能满足以上条件则工序不得插入该时刻,则继续向后遍历寻找该设备的下一个驱动时刻,直至找到满足上述两个条件的时刻插入工序为止。

3 算法和策略的分析与设计

存在预启动设备的综合调度算法,首先需要解决如何减少预启动次数以及提高预启动设备的并行性的问题。结合预启动长路径优先策略和预启动短路径优先策略,提出了预启动路径优先设备驱动调度策略,该策略可以提高预启动设备的并行性。然后对预启动路径优先设备驱动调度策略在调度过程中造成的工序间串行性被破坏问题,提出了可调序列工序调整策略。通过关键路径排序确定工序调整的优先序列,然后按照层级优先对工序进行调整。综上所述本文算法实现流程图如图 1 所示。

图 1 算法实现流程图

3.1 预启动路径优先设备驱动调度策略分析设计

针对存在预启动设备的单车间综合调度问题,本文提出了先根据预启动路径优先的设备驱动策略对加工工序进行初步的排列,为后续的调整策略提供一个初始调度工序序列。该策略的设计步骤如下所述。

(1) 输入设备、工序、加工时间、预启动时间等信息,建立所有工序的约束关系表。

(2) 确定工序表中各工序的状态。

(3) 根据表中的工序以及设备的状态,通过设备状态去寻找此刻可以在该设备上进行的工序,如果设备上没有可以选择的工序则设备继续处于空闲状态,如果存在可加工工序则需要执行步骤(4)。

(4) 如果存在唯一可调度工序则调用该工序,该工序加工完成执行步骤(5),可调度工序不唯一时执行步骤(6)。

(5) 当工序加工完成时,根据设备驱动算法,对新的可调度工序进行选择,并更新工序和设备集中工序和设备状态,执行步骤(8)。

(6) 如果预启动设备处于非加工状态,按预启动长路径优先调度策略对工序进行预调度,并计算出按此策略工序到达预启动设备的时间 t 以及预启动设备开始预启动的时刻,按预启动长路径优先策略对工序进行调度,该工序加工完成执行步骤(5),如果预启动设备处于加工状态则执行步骤(7)。

(7) 预启动设备处于加工状态时,按预启动短路径优先调度策略对工序进行预调度,并计算出按此策略工序到达预启动设备的时间 t 以及预启动设备开始预启动的时刻,按预启动短路径优先策略对工序进行调度,该工序加工完成执行步骤(5)。

(8) 如果工序集中的工序全部加工完毕则执行步骤(9),如果工序集中的工序没有全部加工完毕则返回步骤(3)。

(9) 输出调度结果,工序加工序列以及甘特图。

预启动路径优先设备驱动调度策略实现流程图如图 2 所示。

3.2 可调序列确定策略分析与设计

可调序列确认策略是为了确认目前调度序列中存在可再次进行调整的工序集合,为第 2.4 节中可调序列工序调整策略提供可以调整的目标,对此进行了如下算法设计,该策略的算法流程图如图 3 所示,算法步骤如下所述。

(1) 初次调用可调序列确定策略时的初始加工序列是由预启动路径优先设备驱动调度策略得到的,之后再调用该策略的工序调度序列则为可调序列工序调整策略调整之后的调度序列,输入调度序列以及各设备驱动时刻以及状态和加工时间等信息,继续执行步骤(2)。

(2) 按照工序所需要的加工设备,对工序进行分组,分为 m 组(m 为设备数),每组按调度顺序依次排列,执行步骤(3)。

(3) 确定每台设备最早出现空闲时间段的时刻,并对设备按照时刻升序进行排序。

(4) 确认最早出现空闲时间段的设备,判断该

设备是否调整过, 如果调整过则删除该设备的集合, 返回步骤(3)寻找下一台设备, 如果没有调整过则继续步骤(5)。

(5) 通过步骤(4)中确定的空闲时间段以及设备, 确定该设备在该时刻之后调度的工序集合, 将其放入到一个临时集合中。

(6) 判断临时集合中的工序是否只包含整体加工树的根节点, 如果不是则该集合为可调序列, 然后执行步骤(7), 如果是则假定该设备被调整过。

(7) 得到可调序列。

(8) 结束。

图2 预启动路径优先设备驱动调度策略实现流程图

图3 可调序列确定策略实现流程图

3.3 可调序列工序调整策略分析与设计

为了使调度结果进一步优化设计了可调序列工序调整策略, 对预启动路径优先设备驱动调度策略得到的调度序列进行优化, 使调度结果更加符合经济和生产需求。针对结果中的设备空闲时间段进行调整, 以减少空闲时间段对调度结果造成的影响。

在对调度序列进行调整之前, 首先需要通过可调序列确定策略确认最早出现空闲时间段的设备已经造成影响的序列, 即可调序列确认策略中得到的可调序列, 然后对需要调整的工序进行调整调整。根据第2.4节中对策略思想的描述对策略进行了如下所述的算法步骤设计, 根据以下步骤可调序列工序调整策略实现流程图如图4所示。

(1) 由可调序列确定策略得到本次要调整的序列, 将可调序列中的工序虚拟为根节点, 构成以该节点为根的部分加工树。

(2) 确定上述每棵部分加工树的关键路径并计算每棵部分加工树的关键路径长度, 然后根据每棵树的关键路径长度升序排列可调序列中的工序(每棵部分加工树的虚拟根节点), 构成可调整工序集 Q 。

图4 可调序列工序调整策略实现流程图

(3) Q 中关键路径最短的部分加工树, 按照层级优先的顺序进行排列构成移动工序集 G 。

(4) G 中的第一个工序在其加工设备上寻找符合第2.4节中描述的条件的首个可插入时刻。

(5) 判断工序在插入该时刻之后是否会影响到该设备上该时刻之后加工的后序工序的开始时刻, 如果不影响则执行步骤(7), 如果影响则执行步骤(6)。

(6) 在继续寻找该设备的下一个驱动时刻, 返回步骤(5)。

(7) 该工序插入该时刻, 并更新工序加工序列以及设备加工时间、状态、驱动时刻等信息。

(8) 调整完之后将该工序从 G 中删除, 并更新工序集 G 。

(9) 判断工序集 G 是否为空, 如果不为空返回步骤(4), 如果为空则继续执行步骤(10)。

(10) 将调整完的 Q 中第一个元素从 Q 中删除, 并更新 Q 。

(11) 判断 Q 是否为空, 如果不为空则返回步骤(3), 如果为空则得到调整完成后的工序调度序列以及设备加工信息。

(12) 结束。

4 算法时间复杂度分析

设综合调度问题中产品加工的工序总数为 n , 问题所涉及到的设备总数为 m , 算法的时间复杂度分析如下所述。

(1) 预启动路径优先设备驱动调度策略复杂度的关键在于设备驱动时刻利用预启动路径优先策略对于工序的选择。调度初始时刻所有工序到预启动设备路径长度按快速排序进行排列, 算法的时间复杂度为 $O(n \lg n)$; 设备驱动时刻, 空闲设备寻找可调度工序, 首先需要确认工序是否在该设备上加工, 然后需要对该时刻所有可以在该设备上加工的工序进行比较确认符合条件的工序进行调度, 所以设备驱动时刻最多对所有工序进行两次遍历总次数为 $2n$, 即空闲设备度可调度工序的确认的时间复杂度为 $O(n)$; 设备驱动时刻的计算, 最坏情况为系统中所有设备均处于忙碌状态, 此时需要对所有 m 台设备最多进行 $m-1$ 次比较确认最早出现空闲状态的设备, 问题中总共有 n 道工序, 直至所有工序加工完毕最多进行 $n(m-1)$ 次比较, 所以设备驱动时刻的时间复杂度为 $O(n^2)$ 。综上所述预启动路径优先设备驱动调度策略的时间复杂度为 $O(n^2)$ 。

(2) 每次可调序列的确认, 首先需要对 n 道工序进行一遍遍历将其根据加工设备的不同分成 m 组, 因此将 n 道工序按照设备进行分组的时间复杂度为 $O(n)$; 寻找最早出现空闲时间段的设备, 最坏的情况是对 n 道工序的 n 个设备驱动时刻进行 $n-1$ 次比较, 确认最早出现空闲时间段的设备的时间复杂度为 $O(n)$; m 台设备全部完成可调序列确认需要进行 mn 次比较。综上所述可调序列确认策略的时间复杂度为 $O(n^2)$ 。

(3) 每次可调序列工序调整, 最坏情况是 n 道工序全部在可调序列中, 然后求出每道工序的关键路径并进行排序时间复杂度为 $O(n \lg n)$; 然后对每个虚拟根节点上的工序进行调整, 每道工序调整最多需要对 n 个设备驱动时刻进行 $n-1$ 次比较, 将全部

n 道工序完成调整则需要 $n(n-1)$ 次比较。综上所述可调度工序调整策略时间复杂度为 $O(n^2)$ 。

综上所述, 该算法时间复杂度为 $O(n^2)$ 。

5 实例分析

以上算法的分析和设计过程不以实例为依据, 说明本文提出的策略算法具有普遍意义。为了方便读者理解本文算法, 下面通过实例进一步说明本文算法的执行过程, 并与文献[16]与文献[17]中算法对本实例调度结果进行对比。

设产品 A, 它总共有 19 道工序, 需要在 3 台设备上完成加工, 其中设备 2 为预启动设备(预启动设备不连续加工则需要重新预启动), 图 5 为产品 A 的加工工艺树, 工艺树中每个节点由工序号、设备号和工序加工时间 3 个元素组成, 预启动设备 M2 每次预启动过程的时间为 1。每道工序只有其紧前工序加工完成之后才可以被设备加工, 工序间的紧前紧后关系按照箭头所指方向指示。

图 5 产品 A 加工工艺树

用文献[16]中的算法对产品 A 的调度过程进行分析, 初始 $t=0$ 时刻可调度工序为 {A16, A17, A6, A12, A18, A19, A15, A9}, 此时 M1, M2, M3 均有可调度工序, 设备 2 为预启动设备进行预启动过程。利用最大并行性调度策略选择可调度工序时, 空闲设备 M1, M3 分别选择了 A17, A18, 重复以上过程最终得到甘特图如图 6 所示, 图中阴影块表示预启动设备进行预启动的过程。

图 6 文献[16]算法甘特图(用时 41 工时)

用文献[17]中的算法对产品 A 调度过程, 得到甘特图如图 7 所示。

图 7 文献[17]算法甘特图(用时 48 工时)

如果采用本文算法, 首先需要根据第 3.1 节中的预启动路径优先设备驱动调度策略对产品 A 的所有工序按照该策略进行调度。初始 $t=0$ 时刻可调度工序为 {A16, A17, A6, A12, A18, A19, A15, A9}, 此时 M1, M2, M3 均有可调度工序, 设备 2 为预启动设备进行预启动过程。利用最大并行性调度策略选择可调度工序时, 空闲设备 M1, M3 分别选择了 A16, A19; $t=1$ 时刻设备 M2 处于空闲根据预启动长路径优先策略选择了 A15, 当 $t=5$ 时刻根据预启动短路径优先策略设备 M1 选择了 A17, 如此重复得到表 1 中的数据。

表 1 第 3.1 策略设备驱动时刻表

时刻	空闲设备	M1 选择工序	M2 选择工序	M3 选择工序
$t=0$	M1、M2、M3	A16	预启	A19
$t=1$	M1、	—	A15	—
$t=3$	M3	—	—	A18
$t=5$	M1	A17	—	—
$t=6$	M2	—	A9	—
$t=7$	M3	—	—	A10
$t=10$	M2	—	A12	—
$t=11$	M3	—	—	—
$t=12$	M1、M3	A13	—	—
$t=15$	M2、M3	—	A11	—
$t=16$	M1、M3	A6	—	—
$t=20$	M1、M2、M3、	A14	A7	A5
$t=23$	M1	—	—	—
$t=25$	M1、M2	A3	—	—
$t=27$	M2、M3	—	—	A8
$t=31$	M1、M2	—	—	—
$t=33$	M1、M2、M3	—	—	A4
$t=38$	M1、M2、M3	—	—	A2
$t=41$	M1、M2	—	预启	—
$t=42$	M1、M2、M3	—	A1	—
$t=44$	M1、M2、M3	—	—	—

由表 1 中的数据得到如图 8 所示的甘特图。

图 8 产品 A 执行预启动路径优先设备驱动调度策略甘特图

根据可调序列确定策略确定了, 最先出现空闲时间段的设备为 M3, 得到的调整工序序列为 {A5, A8, A4, A2}; 然后根据可调序列工序调整策略, 分别以这四个节点虚拟为根构成部分加工树, 并分别计算每棵部分加工树的关键路径长度, 并按关键路径长度升序进行排序, 排序结果为 A8, A5, A4, A2。

对以 A8 为根的部分加工树上的工序进行调整, 按照层优先策略对 A8 为根部分加工树树上的工序进行排序, 排序结果为 A19, A14, A15, A8。首先 A19 工序是在 $t=0$ 时刻开始加工的, 不需要进行调整; 其次 A14 工序为 A19 的后序工序需要在 A19 加工完成之后才可以进行调度, 在 A19 完成之后 A14 所在设备 M1 的第一个设备驱动时刻为 $t=5$, 假如将其插入该时刻判断插入之后是否影响设备 M1 在此之后加工的工序的后序工序的开始时间, 如果影响则寻找 M1 下一个驱动时刻, 如果不影响则插入该时刻, 此时 A14 插入 $t=5$ 时刻对其他工序的后序工序不产生影响, 成功插入。如此重复完成对工序 A15 和 A8 的调整。

对 A5、A4 和 A2 重复对 A8 的调整过程, 完成对设备 M3 上可调序列中工序的调整。对设备 M1 和 M2 重复对 M3 的调整过程, 最终得到本文算法结果甘特图, 如图 9 所示。

图 9 本文算法结果图甘特图(用时 36 工时)

本文算法调度结果为 36 工时, 比文献[16]的 41 工时和文献[17]的 48 工时都要短; 本文算法对预启动设备进行了 2 次预启动过程, 而文献[17]则需要进行 3 次预启动。综上所述本文算法在解决存在预启动设备的综合调度问题时, 调度结果更加优化。

6 结论

(1) 通过本文提出的预启动路径优先设备驱动调度策略, 有效地减少了预启动设备的预启动次数, 提高了预启动设备在加工过程中的并行性, 为后序的调整策略提供了一个合理的预调度方案。

(2) 采用本文提出的可调序列确定策略寻找可调序列, 通过调整工序调整策略使可调序列中的工序调整到一个更适当的时刻进行调度, 调度结果更加优化, 并且不影响其他工序的后序工序的调度。

(3) 通过可调序列工序调整策略, 对可调序列中的工序以及调整工序前序进行调整, 对设备驱动破坏的串行关系进行调整, 不仅兼顾了设备并行化, 而且还通过该策略强化了工序间的串行关系。最终使得调度结果进一步优化。

(4) 本文算法时间复杂度比较低, 为 n 的二次方式, 说明本算法简便可行。

参 考 文 献

- [1] 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(6): 191-202.
XIE Zhiqiang, ZHANG Xiaohuan, GAO Yilong, et al. Time-selective integrated scheduling algorithm considering the compactness of serial processes[J]. Journal of Mechanical Engineering, 2018, 54(6): 191-202.
- [2] JOHNSON S M. Optimal two- and three-stage production schedules with setup times included[J]. Naval Research Logistics Quarterly, 1954, 1(1): 61-68.
- [3] LEI D. Pareto archive particle swarm optimization for multi-objective fuzzy job shop scheduling problems[J]. International Journal of Advanced Manufacturing Technology, 2008, 37(1-2): 157-165.
- [4] NAVAIEI J, GHOMI S M T F, JOLAI F, et al. Heuristics for an assembly flow-shop with non-identical assembly machines and sequence dependent setup times to minimize sum of holding and delay costs[J]. Computers & Operations Research, 2014, 44(APR.): 52-65.
- [5] WANG C, JIANG P. Deep neural networks based order completion time prediction by using real-time job shop RFID data[J]. Journal of Intelligent Manufacturing, 2017(4): 1-16.
- [6] FAN H L, XIONG H G, JIANG G Z, et al. Survey of the selection and evaluation for dispatching rules in dynamic

- job shop scheduling problem[C]//Proceedings of the Chinese Automation Congress, 2016: 1926-1931.
- [7] ZHAO F Q, ZOU J H, YANG Y H. A hybrid approach based on artificial neural network(ANN) and differential evolution(DE) for Job-Shop scheduling problem[J]. Applied Mechanics and Materials, 2010, 26-28: 754-757.
- [8] 顾涛, 李苏建, 林莹璐, 等. 周期式退火炉作批处理机的可重入批离散机流水车间调度[J]. 机械工程学报, 2020, 56(2): 220-232.
- GU Tao, LI Sujian, LIN Yinglu, et al. Research on the re-entrant batch discrete flow shop scheduling for periodic annealing furnace as batch processor[J]. Journal of Mechanical Engineering, 2020, 56(2): 220-232.
- [9] 孙在省, 钱斌, 胡蓉, 等. 基于块结构性质的花粉算法求解可重入作业车间调度问题[J]. 机械工程学报, 2019, 55(16): 220-232.
- SUN Zaisheng, QIAN Bin, HU Rong, et al. Flower pollination algorithm based on block structure properties for reentrant Job-Shop scheduling problem[J]. Journal of Mechanical Engineering, 2019, 55(16): 220-232.
- [10] 田云娜, 李冬妮, 郑丹, 等. 一种基于时间窗的多阶段混合流水车间调度方法[J]. 机械工程学报, 2016, 52(16): 185-196.
- TIAN Yunna, LI Dongni, ZHENG Dan, et al. A time window-based approach for multi-stage hybrid flow shop[J]. Journal of Mechanical Engineering, 2016, 52(16): 185-196.
- [11] 赵诗奎. 基于新型邻域结构的混合算法求解作业车间调度[J]. 机械工程学报, 2016, 52(9): 141-150.
- ZHAO Shikui. A hybrid algorithm with a new neighborhood structure for the Job Shop scheduling problem [J]. Journal of Mechanical Engineering, 2016, 52(9): 141-150.
- [12] 赵诗奎. 求解柔性作业车间调度问题的两级邻域搜索混合算法[J]. 机械工程学报, 2015, 51(14): 175-184.
- ZHAO Shikui. Bilevel neighborhood search hybrid algorithm for the flexible Job Shop scheduling problem[J]. Journal of Mechanical Engineering, 2015, 51(14): 175-184.
- [13] 谢志强, 张晓欢, 辛宇, 等. 考虑后续工序的择时综合调度算法[J]. 自动化学报, 2018, 44(2): 344-362.
- XIE Zhiqiang, ZHANG Xiaohuan, XIN Yu, et al. Time-selective integrated scheduling algorithm considering posterior processes[J]. Acta Automatica Sinica, 2018, 44(2): 344-362.
- [14] 谢志强, 刘胜辉, 乔佩利. 基于 ACPM 和 BFSM 的动态 Job-Shop 调度算法[J]. 计算机研究与发展, 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.
- [15] 谢志强, 杨静, 杨光, 等. 可动态生成优先具有优先级工序集的动态 Job-Shop 度算法[J]. 计算机学报, 2008, 31(3): 502-508.
- XIE Zhiqiang, YANG Jing, YANG Guang, et al. Dynamic Job-Shop scheduling algorithm with dynamic set of operation having priority[J]. Chinese Journal of Computers, 2008, 31(3): 502-508.
- [16] 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.
- [17] 谢志强, 辛宇, 杨静. 可回退抢占的设备驱动综合调度算法[J]. 自动化学报, 2011, 37(11): 1332-1343.
- XIE Zhiqiang, XIN Yu, YANG Jing. Machine-driven integrated scheduling algorithm with rollback-preemptive[J]. Automatica Sinica, 2011, 37(11): 1332-1343.
- [18] 谢志强, 桂忠艳, 杨静. 基于设备驱动和实质路径的动态并行综合柔性调度算法[J]. 机械工程学报, 2014, 50(18): 203-212.
- XIE Zhiqiang, GUI Zhongyan, YANG Jing. Dynamic parallel integrated flexible scheduling algorithm based on device driver and essential path [J]. Journal of Mechanical Engineering, 2014, 50(18): 203-212.

作者简介: 谢志强(通信作者), 男, 1962 年出生, 博士, 教授, 博士研究生导师。主要研究方向为企业智能计算与调度优化。

E-mail: xiezhiqiang@hrbust.edu.cn

吕妮, 女, 1992 年出生。主要研究方向为企业智能计算与调度优化。

E-mail: 18233569537@163.com