

存在批量为2的批处理设备的综合调度算法

谢志强^{1,2}, 王悦¹, 杨静²

(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 150001)

摘要: 提出求解存在加工批量为2的批处理设备的复杂单产品综合调度算法。该算法根据复杂产品具有的树状工艺结构的特点,先采用已有的优先级、短用时和长路径策略调度工序;当工序属于批处理设备时,再采用长路径、临界时间等待和先行工序短用时策略。理论分析和实例表明,提出的算法不仅能够解决存在加工批量为2的批处理设备的复杂单产品综合调度优化问题,而且复杂度不超过二次多项式。

关键词: 批处理设备; 复杂产品; 综合调度; 树状工艺结构; 临界时间

中图分类号: TP 278

文献标志码: A

文章编号: 0254-0037(2011)10-1470-07

目前存在批处理设备的调度研究主要解决无约束关系的工件调度问题,其研究经历了不同阶段:单台批加工设备调度、并行批加工设备调度和带有批加工设备的生产线调度^[1-3]。其中,单台批加工设备调度^[4]和并行批加工设备调度主要是基于数学规划、计算智能、离散事件仿真技术和启发式规则,其调度目标是最小化工件总的加权延迟或总的等待成本;带有批加工设备的生产线调度研究主要是基于启发式规则,研究动态批加工设备的加工批量与加工次序^[5]。随着社会对产品需求多样化的增加,多品种、小批量的生产计划越来越多,如果将工件间存在约束关系的复杂单产品加工和装配分别调度,即将产品分解成无约束关系的工件先加工再装配,往往比加工和装配一同处理的综合调度所用的生产时间长^[6-10],因此有必要研究存在批处理设备的复杂产品综合调度问题。

由于复杂产品综合调度过程中经常会遇到批处理设备且无相关研究成果,于是提出存在最基本批处理设备的综合调度算法:存在批处理设备的加工批量为2时复杂单产品的综合调度算法,并用实例验证。

1 问题描述

当存在批处理设备时,复杂单产品调度问题描述为:对于单产品 p ,必须在满足以下3个约束条件的情况下,使得加工产品 p 的时间最短。设产品 p 加工完成需要 n 道工序,即工序 $p_i(i=1,2,\dots,n)$ 。

1) 若满足 $p_j < p_k(j,k=1,2,\dots,n)$, $<$ 为一个偏序关系,即必须等待工序 p_j 加工完毕后才能加工工序 p_k 。

2) 在机器集合 M 中,允许存在批处理设备 M_j ,其加工批量 $N=2$,工序 $p'_i(i=1,2,\dots)$ 是在机器 M_j 上加的工序,在一定条件下,其中的2个并行^[11]工序可以同时加工。

3) 设机器数为 k ,对于机器集合 $M = \{M_1, M_2, \dots, M_i, \dots, M_k\}$,若 p 在 M_i 上加工,则必须等待 p 加工完毕, M_i 方可成为空闲机器。

由于在 $M_i(i=1,2,\dots,k)$ 上加的工序是已知的,设 T_i 为在 M_i 上加工完最后工序的时间,故问题的解可表示为

$$T = \min \{ \max \{ T_1, T_2, \dots, T_i, \dots, T_k \} \}$$

收稿日期: 2009-10-23.

基金项目: 国家自然科学基金项目(60873019,61073043); 黑龙江省自然科学基金项目(F200901); 中国博士后科学基金(20090460880); 黑龙江博士后科学基金(LBH-Z09214); 和哈尔滨市优秀学科带头人项目(2010RFXXG054, 2011RFXXG015)。

作者简介: 谢志强(1962—),男,湖南新邵人,博士后,教授。

2 问题分析与方案设计

一般产品的加工工艺图呈树状结构^[12],只是边的方向与树相反,树上的节点代表工序,边代表偏序关系,叶节点为初始可加工工序,根节点为最后加工的工序.根节点加工完毕时,表明该产品加工完毕.

由加工树和问题的描述可知,初始时只有叶节点工序是被调度的对象,所以初始时只能调度某个叶节点工序.当某个工序的所有子节点工序全部加工完毕时,该工序就转换为叶节点工序.称叶节点工序为可调度工序,所有可调度工序的集合称为备选工序集.

一般调度问题是每种类型的机器只有 1 台,在调度过程中可能存在 2 类设备,即只能处理单一工序的普通设备和批处理设备.当工序在普通设备上加工时,采用文献[13]中的方法调度工序;当工序在批处理设备上加工的情况是本文研究的主要内容,首先要找到所有与该工序并行且可在同设备上加工的其他工序,然后判断是否进行批处理.

2.1 工序在普通设备上加工的情况

当某一可调度工序不在批处理设备上加工时,即在普通设备上加工时,参照文献[13]中的优先级策略、短用时策略和长路径策略进行调度.即优先调度层优先级最高的工序,如果层优先级最高的工序不唯一,则优先调度用时最少的工序;如果用时最少的工序不唯一,则优先调度路径最长的工序;否则,从其中任意选取一个工序.

2.2 工序在批处理设备上加工的情况

当一个可调度工序在某一批处理设备上加工时,如果没有找到与该工序并行且在同设备上加工的其他工序,那么该可调度工序直接进行调度.如果存在与该可调度工序并行且在同设备上加工的其他工序,为了充分利用批处理设备,并使该产品的总调度时间尽可能短,此时需要考虑 2 个问题:1)是否使该可调度工序与找到的其他工序一同调度?2)由于本文只考虑批处理设备的加工批量为 2 的情况,所以当找到的工序中不只一个可以与该可调度工序一同调度,那么选择其中哪个工序与该可调度工序一同加工?针对这 2 个问题,解决方案的设计如下.

问题 1:如果存在与可调度工序 p_i 并行且在同批处理设备上加工的工序 p_j ,由于批处理设备能同时加工 2 个相同工序,那么就要考虑工序 p_j 是否需要等待工序 p_i 一同加工,下面是该情况的分析.

设产品 1 加工树如图 1 所示,其中加工树上的长方块的内容代表:产品工序名/加工机器名/工序加工时间.为方便解释,设工序 p_{im} 与工序 p_{ik} 都在批处理设备 M_1 上加工,时间都是 T ,根节点为 0 层, k, j, m 表示工序的层数, i 表示不同的分支, M_1 的加工批量是 2,其余工序在机器 M_2, \dots, M_n 加工,其中 $1 < k < j < m$.

图 2 是产品 1 中工序 p_{im} 先加工时的甘特图.图 3 是产品 1 中工序 p_{im} 等待工序 p_{ik} 一同加工时的甘特图.由图 2、3,当工序 p_{ik} 的先行工序 $p_{1(k+1)}, p_{1(k+2)}, \dots, p_{1j}$ 所需加工时间等于工序在批处理设备上的加工时间 T 时,2 种情况所用的总工时相同.

当工序 $p_{1(k+1)}, p_{1(k+2)}, \dots, p_{1j}$ 单独或者同时减小时,图 3 所示情况的总工时随之减小;图 2 所示情况的总工时不变,这是因为工序 p_{im} 与工序 p_{ik} 所用时间不变,所以整体时间也不变.由此可得出:当工序 p_{ik} 的先行工序 $p_{1(k+1)}, p_{1(k+2)}, \dots, p_{1j}$ 所需工时小于 T 时,工序 p_{im} 应等待 p_{ik} 之前的工序完成后与 p_{ik} 同时加工.

当工序 $p_{1(k+1)}, p_{1(k+2)}, \dots, p_{1j}$ 单独或者同时增大时,图 2、3 所示情况的总工时相同,此时工序 p_{im} 先加工,不必等待 p_{ik} 之前的工序加工完成后,与 p_{ik} 同时加工.这是因为如果工序 p_{im} 先加工,它的后继工序也会随之提前加工.于是,设 T 为临界时间,如果工序 p_{ik} 的先行工序 $p_{1(k+1)}, p_{1(k+2)}, \dots, p_{1j}$ 所需工时小于 T ,则进行批处理;否则,不必进行批处理.该方法称为临界时间等待策略.

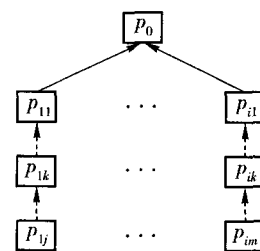


图 1 产品 1 加工树

Fig. 1 Tree model of product 1

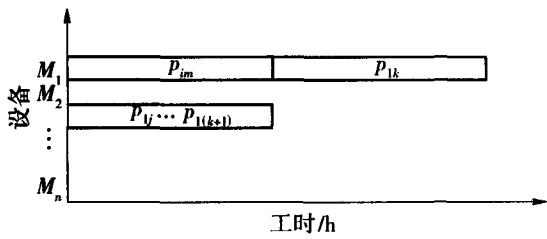


图2 工序 p_{im} 先加工不等工序 p_{ik} 时的甘特图

Fig. 2 Gantt chart of operation p_{ik} not waited by operation p_{im}

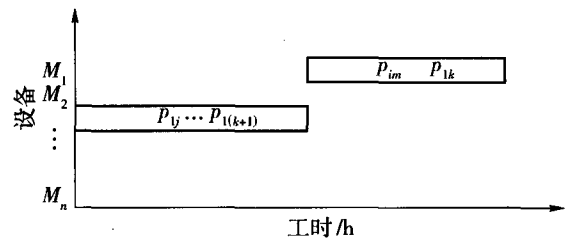


图3 工序 p_{im} 等待工序 p_{ik} 一同加工时的甘特图

Fig. 3 Gantt chart of operation p_{ik} waited by operation p_{im}

例如,有产品2加工树如图4所示,其中机器 M_1 是批处理设备,其加工批量为2.

由于工序 p_2 的先行工序 p_4 所需工时小于工序在批处理设备上加工工时 30 h,那么工序 p_6 要等 p_4 完成后,与 p_2 同时加工.

如果按本策略进行调度,那么调度所用总工时为 70 h,如图5所示.如果在批处理设备上加工工序不等待,那么调度所用总工时为 80 h,如图6所示.可见,采用临界时间等待策略能够充分利用资源且节省时间.

问题2:如果与可调度工序 p_i 在同批处理设备上加工且满足临界时间等待策略条件的并行工序不只一个时,即存在与 p_i 在同设备上加工的并行工序 $p_{i1}, p_{i2}, \dots, p_{ik}$, 并且 $p_{i1}, p_{i2}, \dots, p_{ik}$ 的先行工序所用时间都小于工序在 M_i 上加工的时间 T , 此时采取工序 p_i 优先等待 $p_{i1}, p_{i2}, \dots, p_{ik}$ 之中先行工序用时最短工序的方法. 因为调度用时短的工序,可以使其后继工序获得比较早的可被加工的时间^[14-15]. 该方法称为先行工序短用时策略. 如果在工序 $p_{i1}, p_{i2}, \dots, p_{ik}$ 中,先行工序用时最短的工序不唯一,由于可调度工序的路径越长,对总工时的影响越大;同时优先调度路径较长的工序,并行加工的时间也会较长,而并行加工的时间越长,总工时可能越短^[16]. 所以,此时采用优先调度长路径上工序的长路径策略,根据可调度工序路径的长短来确定调度顺序.

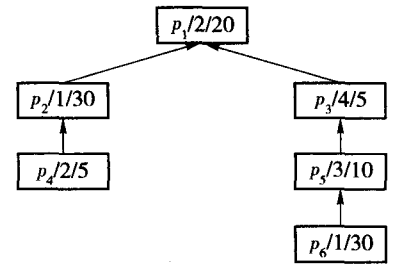


图4 产品2加工树

Fig. 4 Tree model of product 2

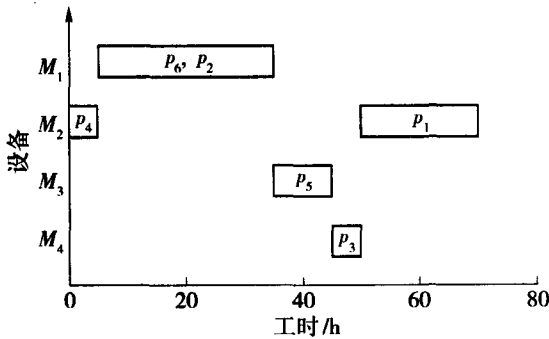


图5 工序 p_6 等待工序 p_4 同时加工时的甘特图

Fig. 5 Gantt chart of operation p_6 waited by operation p_4

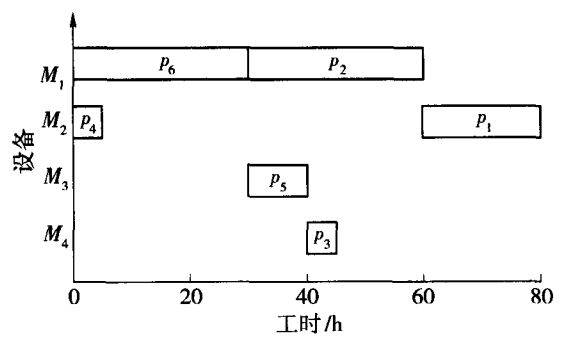


图6 工序 p_6 先加工,不等工序 p_4 时的甘特图

Fig. 6 Gantt chart of operation p_4 not waited by operation p_6

例如有产品3,加工树如图7所示,工序 p_3, p_4, p_5, p_6 都在加工批量为2的批处理设备 M_1 上处理, p_3 的先行工序 p_7, p_4 的先行工序 p_8 和 p_5 的先行工序 p_9 的加工时间都小于工序 p_6 在 M_1 上加工的时间 20 h, 并且 p_9 的加工时间略小于 p_7, p_8 的加工时间. 按照本策略,工序 p_6 应等待先行工序用时较短的 p_5 一同加工,此时所用的总工时是 75 h,如图8所示. 优先调度其先行

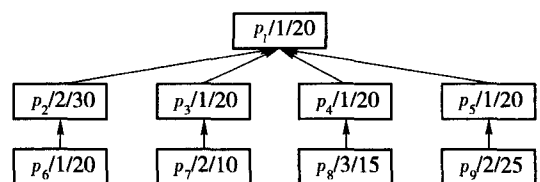


图7 产品3加工树

Fig. 7 Tree model of product 3

工序用时较长的工序时所用的总工时是 85 h, 如图 9 所示. 可见采用先行工序短用时策略能够节省时间.

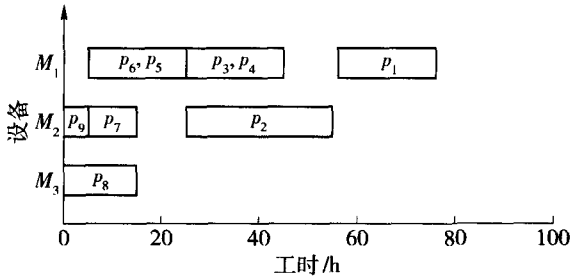


图 8 优先调度短用时先行工序的甘特图

Fig. 8 Gantt chart of scheduling prior to the operation with a shorter time of previous operation

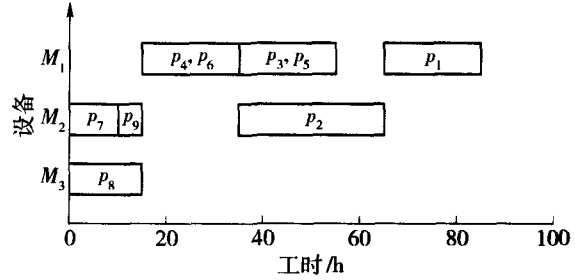


图 9 优先调度长用时先行工序的甘特图

Fig. 9 Gantt chart of scheduling prior to the operation with a longer time of previous operation

对于产品 4, 加工树如图 10 所示, 工序 p_5 、 p_6 、 p_7 都在批处理设备 M_3 上加工, M_3 的加工批量为 2. 根据文献[12]中的优先级策略、短用时策略和长路径策略, 选取工序 p_6 , 然后找到与 p_6 并行且在同设备上加工的工序 p_5 、 p_7 , p_5 、 p_7 没有先行工序, 即它们的先行工序加工时间都等于 0, 小于批处理设备加工时间, 所以按照长路径策略, 选择工序 p_6 与 p_5 一同加工. 优先调度路径较长的 2 个工序 p_6 和 p_5 所用总工时是 120 h (如图 11). 如果优先调度路径较短的 2 个工序 p_4 和 p_3 , 所用总工时则是 130 h (如图 12). 如果优先调度最长路径和最短路径的 2 个工序 p_6 和 p_7 , 所用的总工时则是 125 h (如图 13). 说明采用长路径策略能够节省时间.

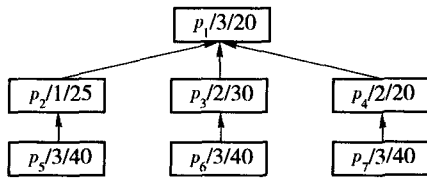


图 10 产品 4 加工树

Fig. 10 Tree model of product 4

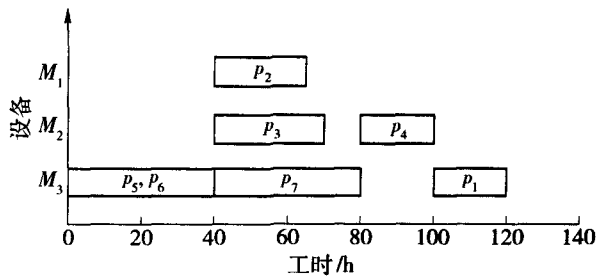


图 11 优先调度长路径工序的甘特图

Fig. 11 Gantt chart of scheduling prior to the operations with a longer path

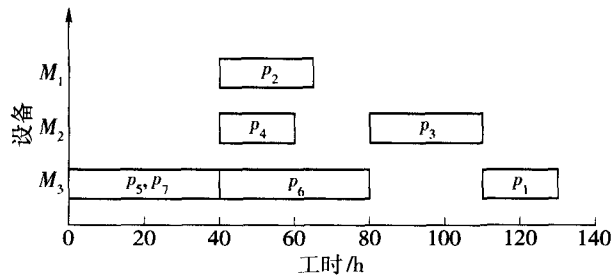


图 12 优先调度短路径工序的甘特图

Fig. 12 Gantt chart of scheduling prior to the operations with a shorter path

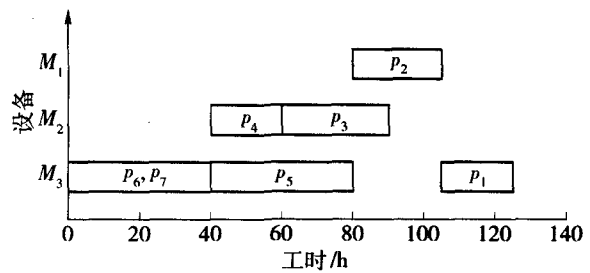


图 13 优先调度最长路径和最短路径工序的甘特图

Fig. 13 Gantt chart of scheduling prior to the operations with the longest path and the shortest path

3 算法设计

- 1) 根据加工树的层结构为工序设置优先级;
- 2) 根据加工树生成备选工序集. 如果工序集为空, 则转到第 8) 步;

3) 从备选工序集中选取层优先级最高的可调度工序,若工序唯一,则选取此工序;若工序不唯一,则选取用时最少的工序;若用时最少的工序不唯一,则选取路径最长的工序;若路径最长的工序不唯一,则在其中任选一个可调度工序;

4) 如果选取的工序不是在批处理设备上加工的,那么调度该工序,然后从加工树上删除该工序所在的节点,从备选工序集中删除该工序,转到第2)步. 如果选取的工序在加工批量为2的批处理设备上加工,则搜索与选取的工序并行且在同批处理设备上加工的其他工序,然后计算搜索到工序的先行工序所需加工时间;

5) 如果搜索到的所有工序的先行工序所需加工时间都大于等于 T ,则不必等待,调度选取的工序;否则,找出其中先行工序用时最少的工序;

6) 如果其先行工序用时最少的工序唯一,则选取路径最长的工序一同调度;

7) 从加工树上删除被调度节点,从备选工序集中删除被调度工序,转到第2)步;

8) 结束.

4 算法复杂度分析

设产品 p 的工序数为 n ,则加工树上有 n 个节点.

1) 采用文献[13]中的优先级策略、短用时策略和长路径策略选出可调度工序,算法复杂度为 $O(n^2)$.

2) 由文献[11],产品加工树中所有的工序结点标记路线号的复杂度为 $O(n)$. 找出与选取的工序并行且在同批处理设备上加工的其他工序,需要比较的次数是 $(n-2) + (n-3) + \dots + 2 + 1 = 0.5n^2 - 1.5n + 1$,算法复杂度是 $O(n^2)$.

3) 计算找到工序的先行工序所需加工时间,然后与批处理时间比较,最坏情况是计算出所有工序的先行工序所需加工时间,共需要计算 n 次,再比较 n 次,所以复杂度为 $O(n)$.

4) 找出先行工序用时最少的工序,最极端的情况是在 $n-1$ 个工序中找出其先行工序用时最少的,需要比较 $n-2$ 次,复杂度为 $O(n)$.

5) 找出路径最长的工序,最极端的情况是计算所有叶节点工序的路径长度,然后找出路径最长的工序,需要计算 $n-1$ 次,比较 $n-2$ 次,复杂度为 $O(n)$.

综上所述,算法复杂度是 $O(n^2)$.

5 实例

采用本文提出的算法(即长路径策略、临界时间等待策略和先行工序短用时策略),对图14所示产品5的加工树进行调度,所需加工工时165 h,调度的结果如图15所示. 采用长路径和先行工序短用时策略,而不采用临界时间等待策略,所需加工工时190 h,调度结果如图16所示. 其中机器 M_1 是加工批量为2的批处理设备. 同理,还可得到其他调度情况所需的总时间,采用各种可能策略所需加工时间如表1所示.

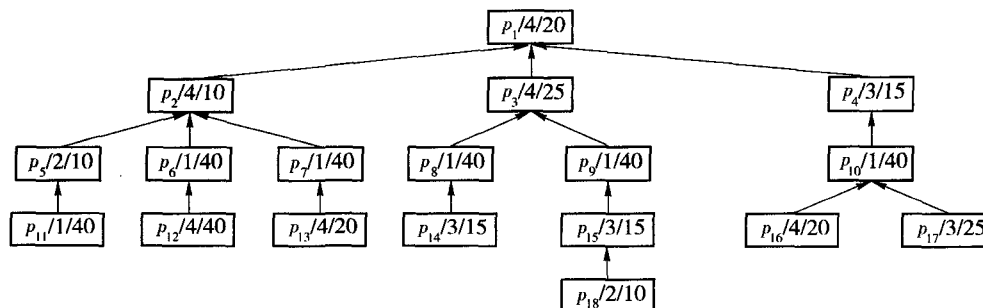


图14 产品5加工树

Fig. 14 Tree model of product 5

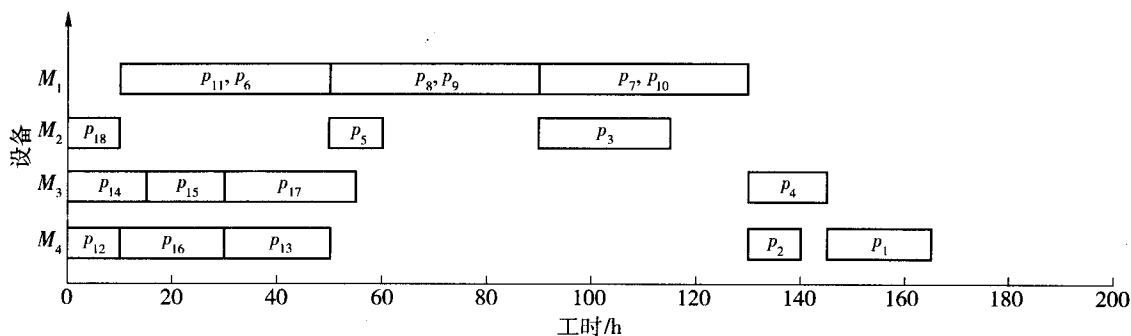


图 15 采用本文算法所得甘特图

Fig. 15 Gantt chart using the algorithm proposed

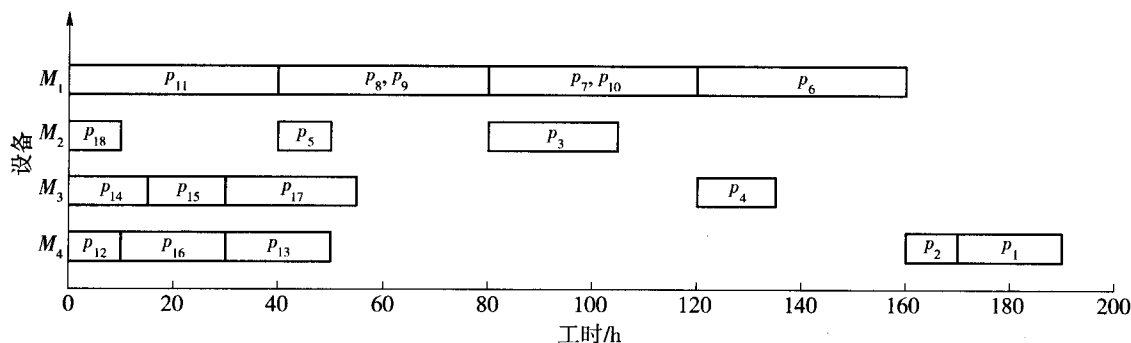


图 16 不采用临界时间等待策略的调度甘特图

Fig. 16 Gantt chart without critical-time waiting strategy

表 1 各种可能策略所需加工时间表

Table 1 Statistics of the possible strategies and process time

采用策略	加工时间/h
临界时间等待策略、长路径策略、先行工序短用时策略	165
临界时间等待策略、长路径策略、先行工序长用时策略	165
临界时间等待策略、短路径策略、先行工序长用时策略	185
临界时间等待策略、短路径策略、先行工序短用时策略	175
不等待策略、长路径策略	190
不等待策略、短路径策略	205

由于先行工序短用时策略是在临界时间等待策略基础上进行的,所以当采用不等待策略时,先行工序短用时策略是不能进行的,即此时只需考虑是否采用长路径策略. 其中,不等待策略是指当在批处理设备上加工的可调度工序数目小于批处理设备的加工批量时,可调度工序不等待其他同设备上工序先行加工.

由此可见,采用本文提出的算法更加节省时间.

6 结论

通过分析以及实例验证,可以看出本文所提出的算法对于加工批量为 2 的批处理设备的综合调度问题,能够得到较优的解. 由于该算法首次提出解决存在批处理设备综合调度问题,因此对深入研究存在批处理设备综合调度问题有一定的借鉴意义.

参考文献:

- [1] LEE C Y, UZSOY R, MARTIN Vega L A. Efficient algorithms for scheduling semiconductor burn-in operations [J]. *Operations Research*, 1992, 40(4): 764-775.
- [2] BRUCKER P, GLADKY A, HOOGEVEEN H, et al. Scheduling a batching machine [J]. *Journal of Scheduling*, 1998, 1(1): 31-54.
- [3] HOCHBAUM D S, LANDY D. Scheduling semiconductor burn-in operation to minimize total flowtime [J]. *Operations Research*, 1997, 45(6): 874-885.
- [4] 牛海军, 孙树栋. JIT 方式下的单机分批调度问题研究 [J]. *西安电子科技大学学报*, 2002, 29(4): 444-460.
NIU Hai-jun, SUN Shu-dong. Research on the batch scheduling on the single machine using the JIT technique [J]. *Journal of Xidian University*, 2002, 29(4): 444-460. (in Chinese)
- [5] UNAL A T, KIRAN A S. Batch Scheduling. *IIE Transaction*, 1992, 20(4): 73-83.
- [6] 谢志强. 工件间有约束的复杂产品工序调度研究 [D]. 哈尔滨: 哈尔滨理工大学机械工程学院, 2009.
XIE Zhi-Qiang. Study on operation scheduling of complex product with constraint among jobs [D]. Harbin: School of Mechanical Engineering, Harbin University of Science and Technology, 2009. (in Chinese)
- [7] XIE Zhi-qiang, YE Guang-jie, ZHANG Da-li, et al. New nonstandard job shop scheduling algorithm [J]. *Chinese Journal of Mechanical Engineering*, 2008, 21(4): 97-100.
- [8] 谢志强, 莫涛, 谭光宇. 非紧密衔接工序动态车间调度算法 [J]. *机械工程学院*, 2008, 44(1): 155-160.
XIE Zhi-qiang, MO Tao, TAN Guang-yu. Dynamic job-shop scheduling algorithm of non-closejoining operations [J]. *Chinese Journal of Mechanical Engineering*, 2008, 44(1): 155-160. (in Chinese)
- [9] 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 Zhi-qiang, LI Zhi-min, HAO Shu-zhen, et al. Study on complex product scheduling problem with no-wait constraint between operations [J]. *Acta Automatica Sinica*, 2009, 35(7): 983-989. (in Chinese)
- [10] XIE Zhi-qiang, HAO Shu-zhen, YE Guang-jie, et al. A new algorithm for complex product flexible scheduling with constraint between jobs [J]. *Computers & Industrial Engineering*, 2009, 57(3): 766-772.
- [11] XIE Zhi-qiang, LIU Yong, HAO Shu-zhen, et al. Study on confirming increasable bottleneck device in complex product dynamic scheduling [J]. *ICCMS*, 2009: 213-217.
- [12] 谢志强, 郑翹, 刘胜辉. 关键设备工序紧凑的动态调度算法 [J]. *哈尔滨理工大学学报*, 2003, 8(2): 50-53.
XIE Zhi-qiang, ZHENG Qiao, LIU Sheng-hui. A dynamic scheduling algorithm based on key equipment's compact procedures [J]. *Journal of Harbin University of Science and Technology*, 2003, 8(2): 50-53. (in Chinese)
- [13] 谢志强, 杨静, 杨光, 等. 可动态生成具有优先级工序集的动态 Job-Shop 调度算法 [J]. *计算机学报*, 2008, 3(31): 502-508.
XIE Zhi-qiang, YANG Jing, YANG Guang, et al. Dynamic job-shop scheduling algorithm with dynamic set of operation having priority [J]. *Journal of Computers*, 2008, 3(31): 502-508. (in Chinese)
- [14] 师瑞峰, 周泓, 上官春霞. 一种求解 job shop 问题的混合多目标遗传算法 [J]. *计算机工程与应用*, 2005(30): 1-5.
SHI Rui-feng, ZHOU Hong, SHANG-GUAN Chun-xia. A hybrid multi-objective genetic algorithm and its applications to job shop problems [J]. *Computer Engineering and Applications*, 2005(30): 1-5. (in Chinese)
- [15] 曾立平, 黄文奇. 求解 Job-Shop 调度问题的一种新的邻域搜索算法 [J]. *计算机研究与发展*, 2005(4): 48-53.
ZENG Li-ping, HUANG Wen-qi. A new local search algorithm for the job shop scheduling problem [J]. *Journal of Computer Research and Development*, 2005(4): 48-53. (in Chinese)
- [16] DROBOUCHEVITCH I G, STRUSEVICH V A. Heuristics for short route job shop scheduling problems [J]. *Mathematical Methods of Operations Research (ZOR)*, 1998(12): 359-375.

Efficient Decoding Algorithm for Unitary Space-time Modulation

SUN En-chang^{1,2}, ZHANG Yan-hua¹, WANG Jie-ling²

(1. School of Electronic Information and Control Engineering, Beijing University of Technology, Beijing 100124, China;

2. State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China)

Abstract: An efficient decoding algorithm for the unitary space-time modulation is presented for the constellations created by a systematic method. First, the receiver calculates the inner products between the received signal and $L/2 + W$ (L is the number of constellations and W is the length of the window) constellations created successively. Second, the $L/2$ sums can be achieved by adding the $L/2 + W$ inner products every W ones, and the maximal one can be searched out. Based on the location of the maximal sum, the searching space of the maximum likelihood (ML) algorithm can be reduced to three constellations. Finally, the optimal constellation from the three constellations can be achieved. The proposed algorithm can significantly reduce the searching space of the ML algorithm. The smaller the value of W , the more the searching space can be reduced. Computer simulation results show that for the unitary constellations, the performance of the proposed algorithm can approach the performance as that of the ML algorithm at signal-to-noise ratio penalty of about 2.0 dB with a half of complexity of ML.

Key words: unitary space-time code; maximum likelihood; searching space

(责任编辑 杨开英)

(上接第 1476 页)

Integrated Scheduling Algorithm With Batch Processing Machines With the Lot-size Being Two

XIE Zhi-qiang^{1,2}, WANG Yue¹, YANG Jing²

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

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

Abstract: A new algorithm is proposed to solve the integrated scheduling problem of single complex product with batch processing machines. The maximum lot-size that the batch processing machines can process is two. According to the feature of the structure of the processing tree that is provided with complex products, the algorithm first adopts the priority strategy, short-time strategy and long-path strategy to schedule operations. When the operations belong to the batching machine, the algorithm then adopts the long-path strategy, critical-time waiting strategy, and short-time of previous operation strategy to schedule operations. Theoretical analysis and examples show that the algorithm can solve the integrated scheduling problem of single complex products with batch processing machines on which the lot-size of operations being processed is two. In addition, the complexity is not more than quadratic polynomial.

Key words: batch processing machine; complex product; integrated scheduling; structure of processing tree; critical-time

(责任编辑 杨开英)