

具有周期维护的最小化加权总完工时间的平行机调度研究

周 菊

贵州大学数学与统计学院, 贵州 贵阳
Email: zj2821367253@163.com

收稿日期: 2021年6月15日; 录用日期: 2021年7月16日; 发布日期: 2021年7月26日

摘 要

针对具有周期维护的最小化加权总完工时间的平行机调度问题, 首先证明了该调度问题是NP-难问题, 并提出最优调度方案的4条性质。然后在WSPT规则的基础上结合工件在多机环境中的分配机制(JCT、MCT、BF)提出了WSPTJCT算法、WSPTMCT算法、WSPTBF算法以及ROPT算法, 其中ROPT算法为其他3个算法的最优输出。最后通过数值实验对4个算法进行性能分析, 结果显示: WSPTJCT算法、WSPTBF算法以及ROPT算法的性能与最大加工时长 p_{\max} 呈倒U型; 工件个数 n 越大, WSPTJCT算法、WSPTMCT算法、WSPTBF算法以及ROPT算法的性能越好。

关键词

周期维护, 平行机调度, WSPT规则, 加权总完工时间

Research on Parallel Machine Scheduling Problem with Periodic Maintenance to Minimize Total Weighted Completion Times

Ju Zhou

Academy of Mathematics and Statistics, Guizhou University, Guiyang Guizhou
Email: zj2821367253@163.com

Received: Jun. 15th, 2021; accepted: Jul. 16th, 2021; published: Jul. 26th, 2021

Abstract

This paper tackles the parallel machine scheduling problem with periodic maintenance to minim-

ize the weighted sum of completion times. This problem is proved to be NP-hard and four properties are proposed. Then, WSPTJCT algorithm, WSPTMCT algorithm and WSPTBF algorithm are proposed with three assignment mechanisms (JCT, MCT, BF) and WSPT rules. ROPT algorithm is the best one of the outputs of the three algorithms. Finally, the performance of the four algorithms is analyzed by numerical experiments. The results show that the performance of WSPTJCT algorithm, WSPTMCT algorithm, WSPTBF algorithm and ROPT algorithm is inverted U-shaped with the maximum processing time p_{\max} and increases with the number of jobs n .

Keywords

Periodic Maintenance, Parallel Machine Scheduling, WSPT Rule, Total Weighted Completion Times

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着社会的发展, 需要加工的工件种类和个数越来越多, 过去不考虑机器维护的短时间加工模型已经无法满足当前需要。因此, 将车间内机器的运行与维护综合考虑是很有必要的, 尤其是周期维护。

在实际生产中, 当货物是小批量生产时, 不同的订单或者说不同的顾客对于企业的重要程度不一定完全相同, 往往需要根据订单或顾客的重要性对每一批订单赋以权重。基于此, 以最小化加权总完工时间为目标函数的调度问题引起了多个领域的注意。具有单次维护的单机调度问题 $1, h_i \parallel \sum w_j C_j$ 是一个强 NP-难问题[1]。可以从最优调度应有的性质的角度对具有维护的该目标函数问题提出相应启发式算法。

从文献角度来看, 目标函数为最小化加权总完工时间的调度问题主要集中于单机环境[2]-[8]。然而, 在实际生产车间中, 机器往往有多台。对于平行机调度问题的目标函数, 研究得更多的是最小化时间表长[9]以及最小化总完工时间[10]。基于此, 我们在具有周期维护的多机环境中思考如何最小化加权总完工时间。

2. 问题描述

现有 n 个在零时刻均已到达可随时加工的独立工件 J_1, J_2, \dots, J_n 需要在 m 台需要进行周期维护的机器 M_1, M_2, \dots, M_m 上加工, 且在加工过程中不可中断。其中工件 J_j 的加工时长为 p_j , 权重为 w_j , 完工时间为 C_j 。所有机器每隔时间单位 T 就需要进行时长为 t 的维护活动, 在进行维护活动时不可加工任何工件。目标是给出工件的完工时刻 C_j 与权重 w_j 乘积之和 $\sum w_j C_j$ 最小的调度方案。当 $p_j > T$ 时, 不存在可行调度方案, 故下文中均假设 $p_j \leq T$ ($1 \leq j \leq n$)。根据三参数表示法[11], 上述调度问题可以被表示为 $P_m, PM \parallel \sum w_j C_j$ 。其中 P_m 表示有 m 台平行机可供使用; PM 表示机器需要周期维护; $\sum w_j C_j$ 表示目标函数为最小化加权总完工时间。

若将机器 M_i 的两个相邻维护之间的时间间隔 T 看作一个时长为 T 的批次, 使用 $B_{i,k}$ 表示机器 M_i 的第 k 个批次, 则调度问题 $P_m, PM \parallel \sum w_j C_j$ 的一个可行调度方案 σ 可以被表示为:

$$\sigma = \{B_{1,1}, t, B_{1,2}, t, \dots, B_{1,k_1}, B_{2,1}, t, \dots, B_{2,k_2}, \dots, B_{m,1}, t, B_{m,2}, t, \dots, B_{m,k_m}\}$$

3. 复杂性分析

由于调度问题 $1, h_i \parallel \sum w_j C_j$ 是 NP-难问题[1], 根据复杂度理论的归结性可知, 具有周期维护的最小

化加权总完工时间的调度问题 $P_m, PM \parallel \sum w_j C_j$ 也是 NP-难问题。

4. 近似算法

4.1. 最优调度方案的性质

由于该调度问题是 NP-难问题, 无法直接给出最优调度方案。但是, 我们可以根据目标函数提出几点最优调度方案应有的性质。

性质 1: 在最优调度方案中, 一定存在 $0 \leq k_{\max} - k_{\min} \leq 1$ 。其中 $k_{\max} = \max\{k_1, k_2, \dots, k_m\}$; $k_{\min} = \min\{k_1, k_2, \dots, k_m\}$ 。

证明 不失一般性, 假设调度方案 σ 是最优调度方案, $k_{\max} = k_1$, $k_{\min} = k_2$, σ 可以被表示为

$$\sigma = \{B_{1,1}, t, B_{1,2}, t, \dots, B_{1,k_1}, B_{2,1}, t, \dots, B_{2,k_2}, \dots, B_{m,1}, t, B_{m,2}, t, \dots, B_{m,k_m}\}$$

其中, $k_{\max} - k_{\min} \geq 2$, 且 B_{1,k_1} 非空, 至少有一个工件 J_j 存在。

若将批次 B_{1,k_1} 内的工件放入 B_{2,k_2} 加工, 其余批次不变, 其目标函数为 f , 则不等式 $f(\sigma) - f \geq w_j(t + p_j) > 0$ 成立。也就是说, 当 $k_{\max} - k_{\min} \geq 2$, 且 B_{1,k_1} 非空时, 还有更优的调度方案。故, 在最优调度方案中, 一定存在 $0 \leq k_{\max} - k_{\min} \leq 1$ 。证毕。

性质 2: 在最优调度方案中, 所有平行机的批次的权重和都应该是由大到小的, 不同机器的批次内的权重和也是。

证明 不失一般性, 假设调度方案 σ_1 在机器 M_i 的调度结果为

$$\sigma_{1,i} = \{B_{i,1}, t, B_{i,2}, t, \dots, B_{i,l}, t, B_{i,l+1}, t, \dots, B_{i,k_i}\},$$

$$\sigma_{1,i+q} = \{B_{i+q,1}, t, B_{i+q,2}, t, \dots, B_{i+q,l}, t, B_{i+q,l+1}, t, \dots, B_{i+q,k_{i+q}}\}。$$

调度方案 σ_2 在机器 M_i 的调度结果为 $\sigma_{2,i} = \{B_{i,1}, t, B_{i,2}, t, \dots, B_{i+q,l+1}, t, B_{i,l+1}, t, \dots, B_{i,k_i}\}$, 在机器 M_{i+q} 的调度结果为 $\sigma_{2,i+q} = \{B_{i+q,1}, t, B_{i+q,2}, t, \dots, B_{i+q,l}, t, B_{i,l}, t, \dots, B_{i+q,k_{i+q}}\}。$

调度方案 σ_1 和 σ_2 除了批次 $B_{i,l}$ 和批次 $B_{i+q,l+1}$ 内的工件发生了对换外, 其余批次没有任何变化。其中, 机器 M_i 的第 l 个批次的权重和为 $\sum_{j=1}^{n_{i,l}} w_j^{i,l}$; 机器 M_{i+q} 的第 $l+1$ 个批次 $B_{i+q,l+1}$ 的权重和为 $\sum_{j=1}^{n_{i+q,l+1}} w_j^{i+q,l+1}$; 机器 M_i 上的调度方案 σ_1 和 σ_2 的目标函数分别为 $f(\sigma_1)$ 和 $f(\sigma_2)$ 。

假设 $\sum_{j=1}^{n_{i,l}} w_j^{i,l} \geq \sum_{j=1}^{n_{i+q,l+1}} w_j^{i+q,l+1}$, 则有

$$\begin{aligned} & f(\sigma_1) - f(\sigma_2) \\ &= \sum_1^{n_{i,l}} w_j^{i,l} C_j^{i,l} + \sum_1^{n_{i,l+1}} w_j^{i,l+1} C_j^{i,l+1} + \sum_1^{n_{i+q,l}} w_j^{i+q,l} C_j^{i+q,l} + \sum_1^{n_{i+q,l+1}} w_j^{i+q,l+1} C_j^{i+q,l+1} \\ &\quad - \sum_1^{n_{i+q,l+1}} w_j^{i+q,l+1} [C_j^{i+q,l+1} - (T+t)] - \sum_1^{n_{i,l+1}} w_j^{i,l+1} C_j^{i,l+1} - \sum_1^{n_{i+q,l}} w_j^{i+q,l} C_j^{i+q,l} - \sum_1^{n_{i,l}} w_j^{i,l} [C_j^{i,l} + (T+t)] \\ &= \sum_1^{n_{i,l}} w_j^{i,l} C_j^{i,l} + \sum_1^{n_{i+q,l+1}} w_j^{i+q,l+1} C_j^{i+q,l+1} - \sum_1^{n_{i,l+1}} w_j^{i,l+1} [C_j^{i+q,l+1} - (T+t)] - \sum_1^{n_{i,l}} w_j^{i,l} [C_j^{i,l} + (T+t)] \\ &= \sum_1^{n_{i+q,l+1}} w_j^{i+q,l+1} (T+t) - \sum_1^{n_{i,l}} w_j^{i,l} (T+t) \\ &= \left(\sum_1^{n_{i+q,l+1}} w_j^{i+q,l+1} - \sum_1^{n_{i,l}} w_j^{i,l} \right) (T+t) \\ &\leq 0. \end{aligned}$$

由以上不等式可知,当批次按照权重和由大到小排列时,其加权总完工时间小于其他调度方案结果。因此,在最优调度方案中,所有平行机的批次的权重和都应该是由大到小的,不同机器的批次的权重和也是。证毕。

性质 3: 在最优调度方案中,任一批次内的工件都是按照 WSPT 规则排序的。

证明 我们知道,同一批次内工件的加工不需要维护。对于无约束的单机调度问题 $\min \sum w_j C_j$, 我们知道 WSPT 规则可以得到最优调度方案。因此,在最优调度方案中,任一批次内的工件都应按照 WSPT 规则排序。证毕。

性质 4: 在最优调度方案中,在加工了工件 J_j 的批次前,所有平行机都没有可以加工工件 J_j 的空闲时间。

证明 不失一般性,假设在调度方案 σ 为最优调度中,工件 J_j 被安排在机器 M_i 的第 $l+1$ 批次 $B_{i,l+1}$ 内加工,完工时刻为 C_j 。且机器 M_{i+q} 的第 l 个批次 $B_{i,l}$ 内有可以加工工件 J_j 的空闲时间。若将工件 J_j 放入批次 $B_{i,l}$ 内加工,其余工件不变,其目标函数为 f , 则不等式 $f(\sigma) - f \geq w_j(t + p_j) > 0$ 成立。故,在最优调度方案中,在加工了工件 J_j 的批次前,所有平行机都没有可以加工工件 J_j 的空闲时间。证毕。

对于单机调度问题 $\min \sum w_j C_j$, 根据 WSPT 规则可以得到最优调度方案。在多机环境中,当有多台机器均可以加工工件时,应该安排在哪一台机器上加工会有更优的结果,我们是不知道的。工件完成时间优先(JCT)、机器完成时间优先(MCT)作为两种工件分配机制被何杰(文献[12])提出用于解决最小化时间表长的车间调度。因此,我们可以将这两个工件分配机制以及 FF 规则与 WSPT 规则结合考虑,提出以下算法。

4.2. WSPTJCT 算法

步骤 1: 将所有工件按照 WSPT 规则排序,不失一般性,假设 $\frac{w_1}{p_1} \geq \frac{w_2}{p_2} \geq \dots \geq \frac{w_n}{p_n}$; 若 $\frac{w_j}{p_j} = \frac{w_{j+q}}{p_{j+q}}$, 且

$p_j \geq p_{j+q}$, 则工件 J_j 排在 J_{j+p} 前;

步骤 2: 若工件 J_j 按照 FF 规则在机器 M_i 上加工的当前完成时刻 C_j^i 不大于在其他机器上加工的完成时刻,则将工件 $J_j (j=1,2,\dots,n)$ 按照 FF 规则安排在机器 M_i 上加工;

步骤 3: 将所有批次按照其权重和由大到小排,依次放入各机器上;

步骤 4: 计算目标值 $f_{WSPTJCT} = \sum w_j C_j$ 。

4.3. WSPTMCT 算法

步骤 1: 将所有工件按照 WSPT 规则排序,不失一般性,假设 $\frac{w_1}{p_1} \geq \frac{w_2}{p_2} \geq \dots \geq \frac{w_n}{p_n}$; 若 $\frac{w_j}{p_j} = \frac{w_{j+q}}{p_{j+q}}$, 且

$p_j \geq p_{j+q}$, 则工件 J_j 排在 J_{j+p} 前;

步骤 2: 若按照 FF 规则加工工件 J_j 后,机器 M_i 的当前完成时刻不大于其他机器加工完工件 J_j 的完成时刻,则将工件 $J_j (j=1,2,\dots,n)$ 按照 FF 规则安排在机器 M_i 上加工;

步骤 3: 将所有批次按照其权重和由大到小排,依次放入各机器上;

步骤 4: 计算目标值 $f_{WSPTMCT} = \sum w_j C_j$ 。

4.4. WSPTBF 算法

步骤 1: 将所有工件按照 WSPT 规则排序,不失一般性,假设 $\frac{w_1}{p_1} \geq \frac{w_2}{p_2} \geq \dots \geq \frac{w_n}{p_n}$; 若 $\frac{w_j}{p_j} = \frac{w_{j+q}}{p_{j+q}}$, 且

$p_j \geq p_{j+q}$, 则工件 J_j 排在 J_{j+p} 前;

步骤 2: 将所有工件按照 BF 规则依次放入各机器上;

步骤 3: 将所有批次按照其权重和由大到小排, 依次放入各机器上;

步骤 4: 不失一般性, 假设时间表长在平行机 M_1 的第 l 个批次内取得, 若每台平行机的使用的批次个数不完全相同, 则将每一台平行机的第 l 个批次内的工件全部取出, 按照 WPSTJCT 算法放在 m 台平行机上加工;

步骤 5: 计算目标值 $f_{WPSTBF} = \sum w_j C_j$ 。

4.5. ROPT 算法

计算目标值 $f_{ROPT} = \min \{f_{WPSTJCT}, f_{WPSTMCT}, f_{WPSTBF}\}$ 。

5. 数值实验

下面我们将通过数值实验来比较 WPSTJCT 算法、WPSTMCT 算法、WPSTBF 算法以及 ROPT 算法的性能的相对优劣。各实验的工件的加工时长均为某时间区间 $[1, p_{\max}]$ 上均匀分布随机生成的正整数, 其中 p_{\max} 为工件最大加工时长。

5.1. $\frac{w_j}{p_j}$ 为任意一组正数

为考察 WPSTJCT 算法、WPSTMCT 算法、WPSTBF 算法以及 ROPT 算法与最大加工时长 p_{\max} 的关系。此数值实验选取了 $p_{\max} = 20, 40, 60, 80, 100$, $n = 400$, $T = 100$, $t = 2$, $m = 2$ 。工件权重 w_i 在区间 $[1, 10]$ 上随机生成。共有 5 组参数, 对不同的参数都随机的生成 100 个实例。

由图 1 可以知道, 当 $n = 400$ 时, 在相同参数下, 这 4 个算法的性能与 p_{\max} 的大小呈倒 U 型, 当 $p_{\max} = 80$ 时, WPSTJCT 算法、WPSTMCT 算法、WPSTBF 算法以及 ROPT 算法的性能最好; ROPT 算法的性能优于 WPSTJCT 算法、WPSTMCT 算法、WPSTBF 算法; WPSTBF 算法优于 WPSTMCT 算法及 WPSTJCT 算法; WPSTMCT 算法优于 WPSTJCT 算法。

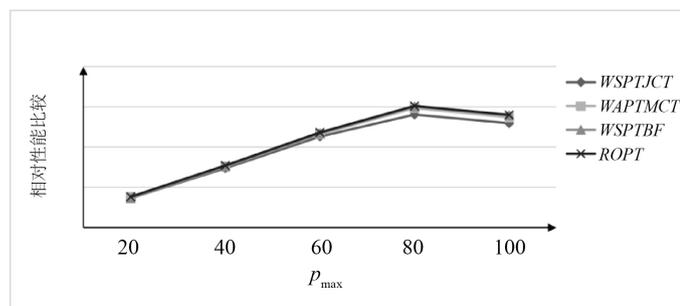


Figure 1. The relationship between performance of algorithms and maximum processing time p_{\max}

图 1. 算法性能与最大加工时长 p_{\max} 的关系

为考察 WPSTJCT 算法、WPSTMCT 算法、WPSTBF 算法以及 ROPT 算法与工件加工个数 n 的关系。此数值实验选取了 $n = 100, 200, 400, 600, 800, 1000, 1500, 2000$ 。 $p_{\max} = 100$, $T = 100$, $t = 2$, $m = 2$ 。工件权重在区间 $[1, 10]$ 上根据均匀分布随机生成。共有 9 组参数, 对不同的参数都随机的生成 100 个实例。

由图 2 可以知道, 在相同参数下: 加工个数 n 越大, WSPTJCT 算法、WSPTMCT 算法、WSPTBF 算法以及 ROPT 算法的性能越好。

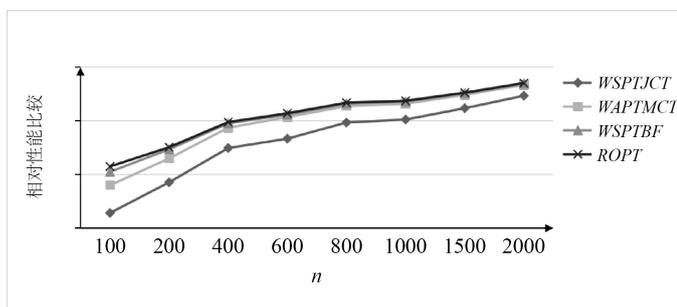


Figure 2. The relationship between performance of algorithms and the number of jobs n

图 2. 算法性能与工件加工个数 n 的关系

相对而言, ROPT 算法的性能优于 WSPTJCT 算法、WSPTMCT 算法、WSPTBF 算法; WSPTBF 算法优于 WSPTMCT 算法及 WSPTJCT 算法; WSPTMCT 算法优于 WSPTJCT 算法; 当 $n \geq 800$ 时, WSPTMCT 算法、WSPTBF 算法以及 ROPT 算法的性能几乎一致, 但 WSPTJCT 算法的性能始终劣于其他三个算法。

5.2. $\frac{w_j}{p_j}$ 为一组单调不减的正数

由于在现实生产车间中存在以工作时长评估订单的重要性的情况, 故在以下实验中假设 $\frac{w_j}{p_j}$ 为一组单调不减的正数, 根据 WSPT 规则可知, 工件是根据 p_j 由大到小排序的。由 WSPTJCT 算法、WSPTMCT 算法以及 WSPTBF 算法的步骤 1 可知, 若 $\frac{w_j}{p_j} = 1$ ($1 \leq j \leq n$), 步骤 1 的排序结果也是根据 p_j 由大到小排序的。因此, $\frac{w_j}{p_j} = 1$ ($1 \leq j \leq n$) 是该情况的一个特例。

为考察 WSPTJCT 算法、WSPTMCT 算法、WSPTBF 算法以及 ROPT 算法与最大加工时长 p_{\max} 的关系。此数值实验选取了 $p_{\max} = 20, 40, 60, 80, 100$, $T = 100$, $t = 2$, $m = 2$, $n = 400$ 。共有 5 组参数, 对不同的参数都随机的生成 100 个实例。

由图 3 可以知道, 当 $n = 400$ 时, 在相同参数下, 其性能变化与图 1 相似, 但其拐点为 60 而非 80。

为考察 WSPTJCT 算法、WSPTMCT 算法、WSPTBF 算法以及 ROPT 算法与工件加工个数 n 的关系。此数值实验选取了 $n = 100, 200, 400, 600, 800, 1000, 1500, 2000$, $T = 100$, $t = 2$, $m = 2$, $p_{\max} = 100$ 。共有 9 组参数, 对不同的参数都随机的生成 100 个实例。

由图 4 可以知道, 在相同参数下其性能变化与图 1 相似, 但存在以下两点不同:

(1) 当 $n \leq 400$ 时, WSPTJCT 算法、WSPTMCT 算法、WSPTBF 算法以及 ROPT 算法的性能变化较快; 当 $n > 400$ 时, 四个算法性能变化较慢;

(2) 相对而言, 图 4 的 WSPTJCT 算法的性能与其他三个算法的性能差没有图 2 的大, 当 $n > 800$ 时, 四个算法性能差别不大, 而图 2 中 WSPTJCT 算法的性能始终小于其他 3 个算法。

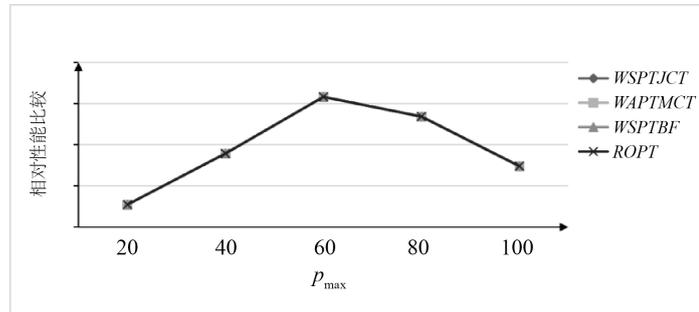


Figure 3. The relationship between performance of algorithms and maximum processing time p_{\max} for a group of non-decreasing positive

numbers $\frac{w_j}{p_j}$

图 3. $\frac{w_j}{p_j}$ 为一组单调不减的正数时, 算法性能与最大加工时长 p_{\max} 的关系

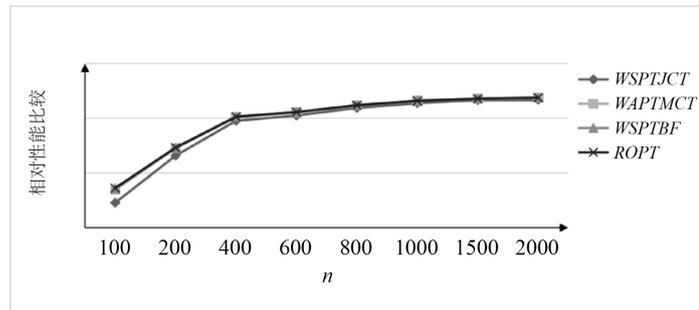


Figure 4. The relationship between performance of algorithms and the

number of jobs n for a group of non-decreasing positive numbers $\frac{w_j}{p_j}$

图 4. $\frac{w_j}{p_j}$ 为一组单调不减的正数时, 算法性能与工件加工个数 n 的关系

6. 结束语

对于具有周期维护, 目标函数为最小化加权总完工时间的平行机调度问题, 提出了 4 个性质, 根据性质以及 3 个工件分配机制提出了 4 个近似算法, 并通过数值实验对 4 个启发式算法进行了性能分析。结果显示: WSPTJCT 算法、WSPTMCT 算法、WSPTBF 算法、ROPT 算法的性能依次增强; 相对而言, 且各算法的性能与最大加工时长 p_{\max} 呈倒 U 型与加工个数 n 呈正相关。从该结果中我们知道 WSPTBF 的算法性能比 WSPTJCT 和 WPSTMCT 算法好, 这提供了一个思路就是将该算法应用于恒速机调度问题中其性能将如何变化, 我们下一步将对此展开研究。

参考文献

- [1] Wang, G., Sun, H. and Chu, C. (2005) Preemptive Scheduling with Availability Constraints to Minimize Total Weighted Completion Times. *Annals of Operations Research*, **133**, 183-192.
<https://doi.org/10.1007/s10479-004-5032-z>

-
- [2] Xu, Z. and Xu, D. (2018) Single-Machine Scheduling with Workload-Dependent Tool Change Durations and Equal Processing Time Jobs to Minimize Total Completion Time. *Journal of Scheduling*, **21**, 461-482. <https://doi.org/10.1007/s10951-017-0543-z>
- [3] Kacem, I. and Chu, C. (2008) Efficient Branch-and-Bound Algorithm for Minimizing the Weighted Sum of Completion Times on a Single Machine with One Availability Constraint. *International Journal of Production Economics*, **112**, 138-150. <https://doi.org/10.1016/j.ijpe.2007.01.013>
- [4] Huo, Y., Reznichenko, B. and Zhao, H. (2012) Minimizing Total Weighted Completion Time with Unexpected Machine Unavailability. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-31770-5_26
- [5] Liu, P., Wang, C. and Lu, X. (2019) A Note on Minimizing Total Weighted Completion Time with an Unexpected Machine Unavailable Interval. *Journal of Scheduling*, **22**, 255-262. <https://doi.org/10.1007/s10951-018-0573-1>
- [6] Krim, H., Benmansour, R., Duvivier, D., et al. (2020) Heuristics for the Single Machine Weighted Sum of Completion Times Scheduling Problem with Periodic Maintenance. *Computational Optimization and Applications: An International Journal*, **75**, 291-320. <https://doi.org/10.1007/s10589-019-00142-5>
- [7] Sadfi, C., Penz, B., Rapine, C., et al. (2005) An Improved Approximation Algorithm for the Single Machine Total Completion Time Scheduling Problem with Availability Constraints. *European Journal of Operational Research*, **161**, 3-10. <https://doi.org/10.1016/j.ejor.2003.08.026>
- [8] Yang, S., Ma, Y., Xu, D., et al. (2011) Minimizing Total Completion Time on a Single Machine with a Flexible Maintenance Activity. *Computers & Operations Research*, **38**, 755-770. <https://doi.org/10.1016/j.cor.2010.09.003>
- [9] 程贞敏, 李洪兴, 谷敏强. 最小化时间表长的平行机调度近似算法研究[J]. 北京师范大学学报(自然科学版), 2012, 48(1): 11-15.
- [10] 曹雁卿. 具有周期维护的最小化工件完成时刻之和的平行机调度问题[J]. 江西科学, 2012, 30(4): 434-437.
- [11] Graham, R.L., Lawler, E.L., Lenstra, J.K., et al. (1979) Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. *Annals of Discrete Mathematics*, **5**, 287-326. [https://doi.org/10.1016/S0167-5060\(08\)70356-X](https://doi.org/10.1016/S0167-5060(08)70356-X)
- [12] 何杰. 预防性维护下的混合型平行机调度问题研究[D]: [博士学位论文]. 长沙: 湖南大学, 2016.