

# 机器带有循环时间窗口的排序问题

曹庭锴<sup>1</sup>, 刘敏<sup>1</sup>, 张同全<sup>2</sup>

<sup>1</sup>云南民族大学数学与计算机科学学院, 云南 昆明

<sup>2</sup>云南民族大学预科教育学院, 云南 昆明

Email: 1481679242@qq.com

收稿日期: 2021年1月3日; 录用日期: 2021年1月30日; 发布日期: 2021年2月8日

## 摘要

给定一个在有限数量机器上加工的作业集合, 如何合理地安排作业在机器上加工以达到最优解就称之为排序问题, 排序问题是经典的组合优化问题之一。机器带有循环时间窗口的排序问题是在我们已知的经典排序问题基础上, 给定机器上的循环时间窗口, 目标是求解机器带有循环时间窗口的排序问题的最小化最大完工时间所用的天数。本文分析了问题的NP困难性, 给出了一种求解机器带有循环时间窗口的排序问题的近似算法, 最后证明了当  $k > m$  时, 算法的最坏情况近似比为  $\frac{3}{2}$ , 当  $k \leq m$  时, 算法具有一个最优平凡解。

## 关键词

NP困难性, 最小化最大完工时间, 近似算法, 循环时间窗口, 排序问题

# The Scheduling Problem with Cyclic Time Windows on Machines

Tingkai Cao<sup>1</sup>, Min Liu<sup>1</sup>, Tongquan Zhang<sup>2</sup>

<sup>1</sup>School of Mathematics and Computer Sciences, Yunnan Minzu University, Kunming Yunnan

<sup>2</sup>School of Preparatory Education, Yunnan Minzu University, Kunming Yunnan

Email: 1481679242@qq.com

Received: Jan. 3<sup>rd</sup>, 2021; accepted: Jan. 30<sup>th</sup>, 2021; published: Feb. 8<sup>th</sup>, 2021

## Abstract

Given a set of jobs which would be processed on a limited number of machines, how to reasonably arrange the processing of jobs on machines to achieve the optimal solution is the scheduling

problem. The scheduling problem with cyclic time windows on machines is based on the classic scheduling problem and given the cyclic time windows on machines, the aim is to minimize makespan for the scheduling problem with cyclic time windows on machines. We analyze the NP-Hardness and present an approximation algorithm for the problem. We prove that the algorithm has a worst-case approximation ratio  $\frac{3}{2}$  when  $k > m$ , and has a trivial optimal solution when  $k \leq m$ .

## Keywords

NP-Hardness, Minimizing Makespan, Approximation Algorithm, Cyclic Time Window, Scheduling Problem

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

机器带有循环时间窗口的排序问题是经典排序问题的一种新的资源约束类型，在经典排序问题中，为达到既定目标，作业需要在一台或者多台机器上进行加工，机器的工作时间没有限制。然而，排序问题的最小化最大完工时间与机器的工作时间有着很大的联系。例如：机器需要定期进行清理与维护，机器的老化，一些复杂的运行程序，作业需要在多台机器上处理等情况。在现实生活中，具有这样情况的排序问题层出不穷，而这类问题算法的好坏关系着资源的利用和生产效率。而在之前的研究文献中，给出的左时间窗口是对于时间点的约束，一旦左端点时间截止，就需要等待一天进行工作。本文所研究的循环时间窗口相较之前的时间窗口即保留其灵活性，又具有时间区间的特性，对于资源利用率的效果更为显著。因此，研究机器带有循环时间窗口的排序问题具有十分重要的意义。

W. E. Smith [1]对单机问题上的总权重完工时间即  $1 \parallel \sum w_j c_j$  问题进行研究，采用相邻的作业进行交换的方法证明了将全部作业遵循权值与其加工时间的比值  $\frac{w_i}{P_i}$  递减的顺序进行作业的加工所得到的排序即是最优解。R. Conway [2]等人对于  $P \parallel \sum C_j$  问题，通过 SPT 法则即按所有作业的加工时间递增的顺序进行加工的方法给出了问题的最优解，而且是多项式内可解的。对于  $P_m \parallel C_{\max}$  问题，R. L. Graham [3]提出了 LS 算法即机器一旦空闲立即加工下一个作业，并且给出算法的最坏情况不超过  $2 - \frac{1}{m}$ 。O. Broun 等人[4]研究了带有时间窗口约束的单机排序问题，给出了对于不同的时间窗口约束的单机排序问题的上界。Yang 等人[5]讨论了带有多个交货期窗口和加工时间可控的排序问题，针对多个目标函数分别给出了多项式时间算法。赵传立等人[6]讨论了带有交货期窗口和加工时间可控的排序问题，证明了在约束条件下是多项式时间内可解的。最后，其他不同约束情况下的排序问题的研究见[7] [8]。

基于上述不同背景情况，本文考虑了一种新的资源约束类型的排序问题即机器带有循环时间窗口的排序问题，分析了问题的 NP 困难性，给出了一种求解机器带有循环时间窗口的排序问题的近似算法，最后证明了当  $k > m$  时，算法的最坏情况近似比为  $\frac{3}{2}$ ，当  $k \leq m$  时，算法具有一个最优平凡解。

## 2. 问题描述

经典的排序问题是以加工完所有作业的最小化的最大完工时间为目标，加工的时间完成的越早，对于资源的消耗越低。但未考虑机器的老化和维修等问题，因此这些在一定程度上也影响着完工时间的大小。我们考虑的排序问题为在带循环时间窗口的情况下，给出一天中机器的工作时间窗口，在机器不工作的时候，对机器进行定时的检查和维修等，对于资源的利用和效率的提高更具有一定的显著性。

现在给出循环时间窗口的定义：机器上的循环时间窗口为机器上每天的统一开始工作和结束工作的时间，即对于每一个  $m_i \in M (i=1,2,\dots,m)$ ，都有对应的循环时间窗口  $[\alpha_1, \alpha_2]$ ， $[\alpha_1, \alpha_2] \subseteq [0, 24]$ ， $\alpha_1$  代表机器统一的开始工作时间， $\alpha_2$  代表机器统一的关闭时间， $p_{j_j} \leq \alpha_2 - \alpha_1$ 。

机器带有循环时间窗口的排序问题描述如下：给出有  $n$  个作业的集合  $J = \{j_1, j_2, \dots, j_n\}$ ，有  $m$  个机器的集合  $M = \{m_1, m_2, \dots, m_m\}$ ；所有作业都在零时刻到达，每台机器不能同时加工两个或者多个作业，作业在加工过程中不能中断。对于每个作业  $j_j \in J (j=1,2,\dots,n)$ ，都有一个相对应的处理时间  $p_{j_j}$ ，所有作业的加工时间的集合记为  $P = \{p_{j_1}, p_{j_2}, \dots, p_{j_n}\}$ 。不同作业之间没有优先级约束，每个作业都必须在  $m$  台机器上分别进行处理，且其处理时间为  $p_{j_j}$ 。如果一个作业的加工时间  $p_{j_j} > \frac{\alpha_2 - \alpha_1}{2}$ ，则此作业称为大作业，否则称为小作业。本文所涉及的作业既有大作业又有小作业。如果一个作业可以被放进一个集合中，则这个集合称为开集，否则就把它记为闭集。如果一个集合变为闭集后，将不能在集合中放置任何作业了。本文所要寻求的目标为机器带有循环时间窗口的排序问题的最小化最大完工时间所使用的天数。

## 3. 研究内容

这一部分是本文研究的主要内容，包括对于机器带循环时间窗口的排序问题的模型建立、算法设计思路、算法设计和算法分析。

### 3.1. 模型建立

$$\begin{aligned}
 Z &= T_{\min} \\
 a_{j_j} &= 0 (j_j \in J (j=1,2,\dots,n)) \\
 p_{j_j} &\geq 0 (j_j \in J (j=1,2,\dots,n)) \\
 x_{j_j, m_i} &\equiv 1 (j_j \in J (j=1,2,\dots,n), m_i \in M (i=1,2,\dots,m)) \\
 s_{j_j, m_i} + p_{j_j} &\equiv t_{j_j, m_i} (j_j \in J (j=1,2,\dots,n), m_i \in M (i=1,2,\dots,m)) \\
 t_{j_j, m_q} &\leq s_{j_j, m_r} (j_j \in J (j=1,2,\dots,n), m_q, m_r \in M (q \neq r, q, r=1,2,\dots,m)) \\
 t_{j_u, m_i} &\leq s_{j_v, m_i} (j_u, j_v \in J (u \neq v, u, v=1,2,\dots,n), m_i \in M (i=1,2,\dots,m)) \\
 p_{j_j} &\leq \alpha_2 - \alpha_1 (j_j \in J (j=1,2,\dots,n)) \\
 m(p_{j_1} + p_{j_2} + \dots + p_{j_n}) &\leq T_{\min} (\alpha_2 - \alpha_1)
 \end{aligned}$$

$a_{j_j}$ ：作业  $j_j$  到达的时间；

$x_{j_j, m_i}$ ：作业  $j_j$  在机器  $m_i$  上加工；

$s_{j_j, m_i}$ ：作业  $j_j$  在机器  $m_i$  上加工开始的时间；

$t_{j_j, m_i}$ ：作业  $j_j$  在机器  $m_i$  上加工结束的时间；

$T_{\min}$ : 所有的作业加工完成所需要的天数。

### 3.2. 算法设计思路

机器带循环时间窗口的排序问题是在经典的排序问题上给定机器每天工作的时间窗口, 以此窗口每天进行循环, 目标是该问题最小化最大完工时间所用的天数。只要保证机器每天多加工作业, 空闲时间尽量减少, 就能更好的提高效率和资源的利用。因为作业在同一时间不能在两台及两台以上机器上加工, 同一台机器不能同时加工两件及两件以上的作业, 因此如何更好的安排作业在机器上的加工就是我们设计算法的关键。

第一, 根据装箱问题的研究, 我们把每台机器的循环时间窗口设为一个个相等容量的“箱子”, 即把每台机器每天能够工作的时间设置为具有加工时间的集合。对于作业按照作业加工时间进行不递减排序, 类比装箱问题的方法, 对于作业依次装入不同的集合中。

第二, 机器带有循环时间窗口的排序问题中所有的作业都要在每天机器上分别加工, 因此对于含有作业的不同集合, 依次在每台机器上进行轮换加工, 既能保证机器不空闲, 又能保证作业在同一时间只在一台机器上加工。因此最后机器带有循环时间窗口的排序问题最小化最大完工时间所用的天数就是和分配的集合的数量相关。

### 3.3. 算法设计

#### 算法 1.1

输入:  $n$  个作业的集合  $J = \{j_1, j_2, \dots, j_n\}$ ,  $m$  个机器的集合  $M = \{m_1, m_2, \dots, m_m\}$ , 每台机器的时间窗口  $[\alpha_1, \alpha_2]$ 。

输出: 最小化最大完工时间所用的天数  $T_{\min}$ 。

Begin

step1: 按照作业的加工时间递减的顺序对作业进行排序, 每个集合可以放置的最多的作业的总加工时长为  $\alpha_2 - \alpha_1$ 。

step2: 把大作业分别放进集合中, 根据放置顺序标记集合, 将这些含有大作业的集合记做“good”集合。小作业的放置顺序如下。

step3: 如果一个最小标记的“good”集合是开集, 且有足够的空间来放置下一个作业, 则把下一个作业放入, 否则这个集合变为闭集。如果集合都为闭集时, 按照如下方法考虑“bad”集合。

(a) 设置一个新的集合为“bad”集合, 若一个“bad”集合为开集且有足够的空间放置作业  $j_j$ , 则把作业  $j_j$  放入这个集合。

(b) 否则“bad”集合成为闭集, 设置一个新的“bad”集放置作业  $j_j$ 。

step4: 如果没有“good”集处于开放状态, 设置一个新的“good”集放置作业  $j_j$ 。因此“good”集的集合记为  $\{G_1, G_2, \dots, G_l\}$ , “bad”集的集合记为  $\{B_1, B_2, \dots, B_{k-l}\}$ 。所以得到

$$D = \{D_1, D_2, \dots, D_k\} = \{G_1, G_2, \dots, G_l, B_1, B_2, \dots, B_{k-l}\}。$$

step5: 若  $k > m$ 。

对于  $i = 1$  到  $m$ 。

则在机器  $m_i$  上按照  $D_i, D_{i+1}, \dots, D_k, D_1, D_2, \dots, D_{i-1}$  的顺序依次加工集合中的作业, 那么  $T_{\min} = k$ 。

若  $k \leq m$ 。

则添加空集  $D_{k+1}, D_{k+2}, \dots, D_m$ , 则在机器  $m_i$  上按照  $D_i, D_{i+1}, \dots, D_k, D_1, D_2, \dots, D_{i-1}$  的顺序依次加工集合中的作业, 那么  $T_{\min} = m$ 。

End

### 3.4. 算法分析

**定理 1.1:** 机器带有循环时间窗口的排序问题是一个 NP 难问题。

证明：我们将装箱问题[9]的实例多项式的转化为机器带有循环时间窗口的实例。

考虑任意一个装箱问题的实例  $I$ ，描述如下：给定  $n$  个物品和  $k$  个容量为  $C$  的箱子， $n$  个物品的体积为  $\{s_1, s_2, \dots, s_n\}$ ，然后将  $n$  个物品分别放入箱子中，问题是  $k$  个箱子能装下所有的物品吗？

我们将装箱问题的实例多项式的转化为机器带有循环时间窗口的实例  $I'$ ，描述如下：给定  $n$  个作业的集合和一台机器，这台机器上的循环时间窗口为  $[\alpha_1, \alpha_2]$ ，即这台机器一天的工作时间为  $C = \alpha_2 - \alpha_1$ 。所有作业处理时间的集合为  $\{s_1, s_2, \dots, s_n\}$ ，然后  $n$  个作业分别在这台机器上进行加工，使得所有作业加工完成的最小化最大完工时间的天数为  $k$ 。

接下来，我们证明了如下结论：装箱问题的实例  $I$  答案为“是”当且仅当机器带有循环时间窗口的排序问题的实例  $I'$  具有  $k$  值的最优解。

如果装箱问题的实例  $I$  答案为“是”，那么我们假设实例  $I$  有一个装配方案  $X = \{X_1, X_2, \dots, X_k\}$ 。对于机器带有循环时间窗口的排序问题，将处理时间等于  $X_i$  中物品体积的作业放入同样的集合中，因此我们就可以得到一个好的加工排序作业的集合  $X = \{X_1, X_2, \dots, X_k\}$ 。然后在机器上分别加工这些子集中的作业，每个作业子集可能需要一天的时间来处理，所以使用的最小化最大完工时间的天数为  $k$ 。因为装箱问题的实例  $I$  所使用的箱子数为  $k$ ，因此对于实例  $I'$  我们有一个最优解，且其最优目标函数值为  $k$ 。

对于机器带有循环时间窗口的排序问题的实例  $I'$ ，如果存在一个目标函数值为  $k$  的最优解，然后把加工作业的子集的集合记为  $X = \{X_1, X_2, \dots, X_k\}$ ，将体积等于  $X_i$  中作业处理时间的物品放入同样的集合中，因此我们就可以得到一个好的包含所有物品的箱子的集合  $X = \{X_1, X_2, \dots, X_k\}$ ，且其所使用的箱子数为  $k$ ，因此我们可以得到装箱问题的实例  $I$  的答案为“是”。

但是装箱问题是一个 NP 难问题[9]，那么机器带有循环时间窗口的排序问题同样也属于 NP 难问题。

**定理 1.2:**  $k > m$  时，算法 1.1 的最坏情况近似比为  $\frac{3}{2}$ ，当  $k \leq m$  时，算法具有一个最优平凡解。

证明：对于机器带有循环时间窗口的排序问题，用  $A$  表示算法 1.1 所得到的结果，用  $OPT$  表示最优的结果。由算法 1.1 得到的子集是由两个不同的集合  $\{G_1, G_2, \dots, G_l\}$  和  $\{B_1, B_2, \dots, B_{k-l}\}$  组成的， $\{G_1, G_2, \dots, G_l\}$  代表着所有的“good”集， $\{B_1, B_2, \dots, B_{k-l}\}$  代表着所有的“bad”集。我们把关闭的“good”集的数量记为  $c$ ，当每一个“good”集关闭时，意味着有一个作业被放进一个“bad”集；反之，当有一个作业被放进一个“bad”集时，一个“good”集关闭。因此“bad”集中所包含的总的作业数目为  $c$ 。因为“bad”集中的作业均为小作业，所以除了最后一个“bad”集可能包含一个小作业，其余“bad”集至少包含两个小作业，因此，能够得到  $k - l \leq \left\lfloor \frac{c}{2} \right\rfloor$ ，当  $k > m$  时，存在两种情况：

情形 1：当  $c \leq l - 1$  时

我们很容易得到：

$$A = k = l + k - l \leq l + \left\lfloor \frac{c}{2} \right\rfloor \leq l + \left\lfloor \frac{l-1}{2} \right\rfloor \leq \frac{3}{2}l$$

接下来我们讨论  $OPT$  的结果：

当  $c = l - 1$  时， $OPT \geq \frac{\sum p_i}{\alpha_2 - \alpha_1} = \frac{\sum_{i_j \in G_i} p_i}{\alpha_2 - \alpha_1} + \frac{\sum_{i_j \in B_i} p_i}{\alpha_2 - \alpha_1} > c = l - 1$ 。当  $c \leq l - 2$  时，每一个“good”集都有一个大的作业，那么上式也成立。

因此当  $c \leq l - 1$  时，我们可以得到  $\frac{A}{OPT} \leq \frac{3}{2}$ 。

情形 2: 当  $c = l$  时

$$A = k = l + k - l \leq l + \left\lceil \frac{c}{2} \right\rceil \leq l + \left\lceil \frac{l}{2} \right\rceil \leq \frac{3}{2}l + \frac{1}{2}$$

$$OPT > c = l$$

因此当  $c = l$  时,  $\frac{A}{OPT} \leq \frac{3}{2}$  也成立。

对于  $k \leq m$  的情况, 因为所有作业中至少有一个大作业, 所以大作业在每台机器上加工都需要一天, 所以  $k \leq m$  时, 最小化最大完工时间所需要的天数为  $m$ , 它是一个平凡最优解。

对于算法 1.1 通过上述证明过程, 我们能够得到当  $k > m$  时, 算法 1.1 的最坏情况近似比为  $\frac{3}{2}$ , 当  $k \leq m$  时, 算法具有一个最优平凡解。

**定理 1.3:** 算法 1.1 的时间复杂度为  $O(n \log n)$ 。

证明: 算法中的 step1 能够在  $O(n \log n)$  时间范围内完成, step5 的计算量为  $O(m)$ , 其他步骤的时间复杂度为  $O(n)$ , 所以算法 1.1 的时间复杂度为  $O(n \log n)$ 。

#### 4. 结束语

本文通过研究带有循环时间窗口的一类排序问题, 给出机器上的循环时间窗口, 定义一种新的排序问题的约束类型, 设计其问题的算法, 并分析算法的时间复杂度。这一算法对于机器需要定期进行清理与维护, 机器的老化, 一些复杂的运行程序, 作业需要在多台机器上处理等情况的排序问题具有重要的应用价值, 能够更好地提高资源的利用率和效率。

#### 致 谢

非常感谢对本稿提出意见的审稿人及老师。

#### 参考文献

- [1] Smith, W. (1956) Various Optimizers for Single-Stage Production. *Naval Research Logistics Quarterly*, **3**, 59-66. <https://doi.org/10.1002/nav.3800030106>
- [2] Conway, R., Maxwell, W. and Miller, L. (1967) *Theory of Scheduling*. Addison-Wesley, Hoboken.
- [3] Graham, R. (1966) Bounds for Certain Multiprocessor Anomalies. *Bell System Technical Journal*, **45**, 1563-1581. <https://doi.org/10.1002/j.1538-7305.1966.tb01709.x>
- [4] Braun, O., Chung, F. and Graham, R. (2013) Single-Processor Scheduling with Time Restrictions. *Journal of Scheduling*, **17**, 399-403. <https://doi.org/10.1007/s10951-013-0342-0>
- [5] Yang, D.L., Lai, C.J. and Yang, S.J. (2014) Scheduling Problems with Multiple due Windows Assignment and Controllable Processing Times on a Single Machine. *International Journal of Production Economics*, **150**, 96-103. <https://doi.org/10.1016/j.ijpe.2013.12.021>
- [6] Zhao, C.L. and Tang, H.Y. (2012) A Note to Due-Window Assignment and Single Machine Scheduling with Deteriorating Jobs and a Rate-Modifying Activity. *Computers & Operations Research*, **39**, 1300-1303. <https://doi.org/10.1016/j.cor.2010.04.006>
- [7] Zhang, Y., Dan, Y.R., Dan, B., et al. (2019) The Order Scheduling Problem of Product-Service System with Time Windows. *Computer and Industrial Engineering*, **133**, 253-266. <https://doi.org/10.1016/j.cie.2019.04.055>
- [8] Mosheiov, G., Sarig, A., Strusevich, V.A., et al. (2018) Two-Machine Flow Shop and Open Shop Scheduling Problems with a Single Maintenance Window. *European Journal of Operational Research*, **271**, 388-400. <https://doi.org/10.1016/j.ejor.2018.04.019>
- [9] Zhang, G.C., Cai, X.Q. and Wong, C.K. (2000) Linear Time-Approximation Algorithms for Bin Packing. *Operation Research Letters*, **26**, 217-222. [https://doi.org/10.1016/S0167-6377\(99\)00077-2](https://doi.org/10.1016/S0167-6377(99)00077-2)