Parallel Machines Parts Ordering Problem with Time Window

Bihui Cheng*

School of Yunnan University of Nationalities, Kunming 650500 China

*Corresponding author e-mail: 383302216@qq.com

Abstract: in the classical sorting problem, the processing time of the job is usually a constant independent of the start time. However, in some sorting problems with strong practical background, the processing time of the job may have some relation with the starting time. It is divided into two cases, one is the processing time of the job is the increase function of the start time, the other is the processing time of the job is the decrease function of the start time. On the only one machine sorting problem with in time windows, this paper designs a 3/2-approximation algorithm for the parallel machiness parts sorting problem with in time windows, and its algorithm complexity is.

1. Problem description

Compared with the sorting problem of only one machine job with in time windows, the sorting problem of parallel machines job with in time windows is still defined as the in time windows of each machine\([a, b]\), for any job in any machine processing time is \(t\), And the number of machines is \(m\). We still require that a job must be finished at the beginning of processing, and no interruption is allowed.

**Problem 1** Given an artifact \(J_1, J_2, J_3, \ldots, J_n\) and \(m\) Machine, Among them \(a\) represents the startup time of the machine, \(b\) is the shutdown time of the machine, \(a, b \in [0, 24)\), \(t\) represents the processed job \(J_i\) time required. Definition \(s_i = (x_i, y_i, z_i)\) said the job \(J_i\) in the first \(x_i\) day \(y_i\) Moment in \(Z_i\) top machining, for any job \(J_i\) Satisfy the inequality \(y_i + t_i \leq b\). For the machine \(M\) The

---

* Project status: funded by the scientific research fund of the education department of yunnan province
Cheng bihui (1989-), Dali, yunnan, lecturer, research direction: combinatorial optimization
The total number of days processed is recorded as \( D_j \). The goal is to maximize \( \min(D_j) \).

**Theorem 1** Ordering of parallel machines parts with in time windows is a \( \text{NP-hard} \) problem.

**Proof:** Because of theorem 1, the number of parallel machines in the sorting problem of parallel machines parts with in time windows is required to be 1, then the sorting problem of parallel machines parts with in time windows is transformed into the sorting problem of only one machine parts with in time windows \( P \neq \text{NP} \). Under the premise that the hypothesis is true, the only one machine job ordering problem with in time windows is \( \text{NP-hard} \).

So the parallel machines parts ordering problem with in time windows yes.

Because of theorem 1, we know that there is no polynomial optimal algorithm for sorting parallel machines parts with in time windows, so we design a 2-approximation algorithm here.

The following is the specific idea to solve the sorting problem of parallel machines parts with in time windows. The specific strategy of our design method is as follows:

1. In the sorting problem of parallel machines parts with in time windows, the number of parallel machines is required to be 1.
2. Call the algorithm and get a feasible solution.
3. The feasible solution is generalized to the machine, and the startup time of each job is calculated.
4. Total days of output processing and start time of each job.

2. Algorithm design

described as follows designed is:

**Algorithm 1:** DSPWS

Input: for a given \( n \) a job \( J_1, J_2, J_3, \ldots, J_n \) and the machine \( M \), the in time windows is \([a, b]\), artifacts \( J_i \) Processing time is \( t_i \).

Output: the number of days of job processing \( d \) and the startup time of each job \( \{s_1, s_2, \ldots, s_n\} \).

Begin

Step 1 To calculate \( \tau = b - a \) according to processing time \( t \), from big to small Step 2 call quick sort algorithm, according to the processing time, will all artifacts from big to small new arrangement. Row the column order is written as \( J'_1, J'_2, J'_3, \ldots, J'_n \). Let \( i = 1, \quad d_i = 0, \quad 1 \leq i \leq n \).

Step 3 for the job \( J'_i \), the original processing order is denoted as \( k \), so \( J'_i = J_k \).

Step 4 If \( t_k + d_j \leq \tau \), has made \( x_k = j, \quad y_k = a + d_j \) and order \( s_i = (x_i, y_i) \) make \( d_j = t_k + d_j \), if \( i < n \), has made \( i = i + 1 \), go to the Step 3 Otherwise, go to the Step 5; if \( t_k + d_j > \tau \), make \( j = j + 1 \), go to the Step 4.

Step 5 make \( d = \max \{s_i\} \). total days \( d \) and each output machining job boot time \( \{s_1, s_2, \ldots, s_n\} \).
End

**Algorithm 2 DDPWS**

Input: Given an artifact $J_1, J_2, J_3 \ldots J_n$ and $m$ machine $M$, The in time windows is $[a, b]$, artifacts $J_i$

processing time is $t_i$.

Output: the number of days of job processing $d$ and the startup time of each job $\{s_1, s_2, \cdots s_n\}$.

Begin

Step 1 make $m' = m$, $m = 1$. Call the algorithm DSPWS, get $d'$ And the startup time of each job $\{s'_1, s'_2, \cdots s'_n\}$

$$s = \left\lfloor \frac{d'}{m'} \right\rfloor$$

Step 2 make $i = 1$ to calculate

Step 3 for $s'_i = (x_i, y_i)$, Find the corresponding two non-negative integers $p$ and $q$, meet $x_i = p \cdot d + q$, in its $0 < q \leq d$.

Step 4 and $x_i = p + 1$, $z_i = q$ .make $s_i = \{x_i, y_i, z_i\}$, make $i = i + 1$, Go to the Step 3

Step 5 the days $d$ and each output machining job boot time $\{s_1, s_2, \cdots s_n\}$

Then the algorithm is a 2-approximation algorithm to solve the sorting problem of only one machine and job with time-interval window, and its time complexity is $O(n \log n)$.

Proof: for convenience and clarity of proof, the minimum total number of days of the job processed for an input instance is defined as $OPT$, Corresponding to $DDPWS$ the total number of days processed by the algorithm is defined as $OUT$.

According to the sorting problem limitation, a job cannot be interrupted once it is machined, and we require it to be machined at the beginning it's later than that $a$, The finish time of the job processing should be earlier than the time $b$. So algorithmically Step 3 and Step 4.

For any instance, let's say $m = 1$ the optimal solution is denoted as $d^*$, There are clearly

And the algorithm $DDPWS$ The output solution satisfies:

$$d = \left\lfloor \frac{d'}{m} \right\rfloor$$

Because of the theorem, we have

$$d' \leq \frac{3}{2}d^* + 1$$
From that, we get \[ \text{OUT} = d = \left\lceil \frac{d''}{m'} \right\rceil \leq \left\lfloor \frac{3}{2} \cdot \frac{d'' + 1}{m'} \right\rfloor \leq \left\lfloor \frac{3}{2} \cdot \frac{d'}{m'} \right\rfloor + 1 = 2 \cdot \text{OPT} + 1 \]

The algorithm DDPWS is an approximate algorithm with an approximate value of 2-

The following analysis DDPWS Algorithm time complexity:

We know that the algorithm complexity mainly comes from the Step1 call DSPWS algorithm, DSPWS Algorithm, its time complexity is \( O(n \log n) \).

3. Conclusion

By studying the problem of only one machine with in time windows, the polynomial algorithm of in time windows sorting problem in parallel machines is designed. In view of the problems to be solved in this paper, more reasonable steps are designed to reduce the time complexity.

References

[1] Optimal two and three stage production schedules with setup times included, Naval Research Logistics Quarterly, 1954(1):61-68.
[2] tang hengyong, zhao chuanli. Introduction to sorting [M]. Beijing: science press, 2002.1-9
[3] P.B rucker. The Scheduling Algorithms, Springer, 2007
[4] b. Chen, C.N. Potts, Geraldine Jim Woeginger. A review of the machine scheduling: the and approximability, faced of combinatorial optimization (D.Z. Du and P.M. Pardalos eds.), Kluwer Academic Publishers, 1998.
[5] Journal of shenyang normal university (natural science edition), 2014,32(04):453-460.
[6] Yu ying. A sort of sorting problem related to processing time and location [J]. Journal of kaili university, 2011,29(03):17-19.
[7] Zhao chuanli, zhang lei. Scheduling problem with delivery window and controllable processing time [J]. Journal of shenyang normal university (natural science edition), 2016,34(04):402-408.
[8] Zhong weiya, ma xiaoru. Scheduling problem of no waiting assembly line with transport and processing flexibility [J]. Journal of operations research, 2016,20(04):93-101.
[9] tang guochun. Research on the scheduling problem of missed work [J]. Journal of chongqing normal university (natural science edition), 2009,26(02):1-6+17.
[10] sun shijie. Scheduling problems in just-in-time production and tolerant delivery [J]. Mathematical theory and application, 1999(03):69-75.