Scheduling with periodic availability constraints to minimize makespan

Lishi Yu, Zhiyi Tan

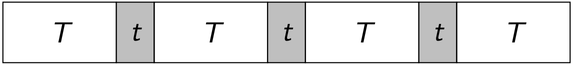
Department of Mathematics
Zhejiang University

July 5, 2021

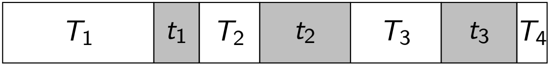
Catalogue

- Introduction
- Related Work
- Lemma
- $1|nr-pm|C_{max}$
- $P2|nr-pm|C_{max}$
- Conclusion

Introduction



An illustration of **periodic unavailability periods**. Here, T denotes the length of each available period, t denotes the length of each unavailable period.



An illustration of **random unavailability periods**. Here, T_i denotes the length of ith available period, t_i denotes the length of ith unavailable period.

Introduction

- Given a set of n independent jobs $\mathcal{J} = \{J_1, \ldots, J_n\}$, which are to be processed on $m \geq 1$ parallel identical machines $M_1, M_2, \ldots, M_m$. The processing time of J_j is p_j, $j = 1, \ldots, n$.

- All the jobs are available at time zero, and no preemption is allowed. Each machine is periodically unavailable.

- The duration of each unavailable period and available period is t and T, respectively. Denote by $\beta = \frac{t}{T}$ the ratio between the length of an unavailable period and available period. In most realities, $\beta < 1$.

Introduction

- Without loss of generality, we assume that each machine just finishes its maintenance at time 0 and $T \geq p_1 \geq p_2 \geq \ldots \geq p_n$.

- The objective is to minimize the makespan, which is the maximum completion time among all machines.

Introduction

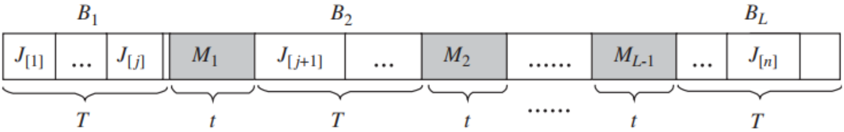


Figure: An illustration of the problem under consideration, take the case of m=1 as an example, where $J_{[j]}$ denotes the job placed in the jth position of the given schedule.

By extending the three-field notation, the problem was denoted as $Pm|nr-pm|C_{max}$ in Ji, He, and T. Cheng, 2007. Let C^A be the objective value of the algorithm solution and C^* be the objective value of the optimal solution, $\beta = \frac{t}{T}$.

Related Work

There are plenty of research on scheduling with unavailability periods. We refer to Lee, 2004 and Ma, Chu, and Zuo, 2010 for the surveys on this topic.

- $1|nr - pm|C_{max}$: Ji, He, and T. Cheng, 2007 proposed an algorithm LPT, and showed that its worst-case ratio is 2. Moreover, there is no polynomial time approximation algorithm with a worst-case ratio of less than 2 unless $P = NP$.

- $P2|nr - pm|C_{max}$: Sun and H. Li, 2010 introduced an algorithm and proved that its worst-case ratio is at least $\max\left\{\frac{14}{11} + \frac{12}{11}\beta, 2\right\}$ and at most $\max\left\{\frac{8}{5} + \frac{6}{5}\beta, 2\right\}$.

Related Work

- Results on corresponding problems with different objectives and other variations can be found in Qi, Chen, and Tu, 1999; Qi, 2007; Xu, Z. Cheng, et al., 2009; Xu, Yin, and H. Li, 2009; Xu, Sun, and H. Li, 2008; Sun and H. Li, 2010; G. Li and Lu, 2015; Gawiejnowicz, 2020a; Gawiejnowicz, 2020b.

Related Work

Algorithm *Longest Processing Time first* (*LPT* for short)

First sorts the jobs in non-increasing order by processing times, then always assigns the first unprocessed job in the sequence to the machine which can complete it as early as possible.

Related Work

Why we study $1|nr-pm|C_{max}$?

• Ji, He, and T. Cheng, 2007: Both the tightness of LPT and the non-approximability only valid when β tends to infinity, which falls into the relatively unrealistic situation.

• Ji, He, and T. Cheng, 2007: The performance of LPT when β is small remains unexplored.

• Yu, Zhang, and Steiner, 2014: presented worst-case ratios of LPT algorithms based on other bin-packing algorithms as functions of b^*, the minimum number of availability periods that at least one job is processed on in any schedule. However, the parameter b^* is instance-dependent and it is NP-hard to obtain its exact value.

Our Results(m=1)

Theorem 1

The worst-case ratio of the LPT for $1|nr-pm|C_{max}$ is no more than

$$r(\beta)=\begin{cases}\frac{44+44\beta}{33+36\beta}, & \beta\in(0,\frac{\sqrt{313}-15}{32}]\approx(0,0.0841],\\ \frac{29+28\beta}{22+20\beta}, & \beta\in(\frac{\sqrt{313}-15}{32},\frac{\sqrt{181}-11}{24}]\approx(0.0841,0.1022],\\ \frac{9+8\beta}{7+4\beta}, & \beta\in(\frac{\sqrt{181}-11}{24},\infty)\approx(0.1022,\infty),\end{cases}$$

and the bound is tight when $\beta\geq 0.1022$.

Our Results(m=2)

Theorem 2

The worst-case ratio of the DFFD for $P2|nr-pm|C_{max}$ is $\frac{10}{7}+\frac{8}{7}\beta$, and the bound is tight.

Theorem 3

For $P2|nr-pm|C_{max}$, there is no polynomial time approximation algorithm with a worst-case ratio of less than $1+\beta$ unless $P=NP$.

Our Results

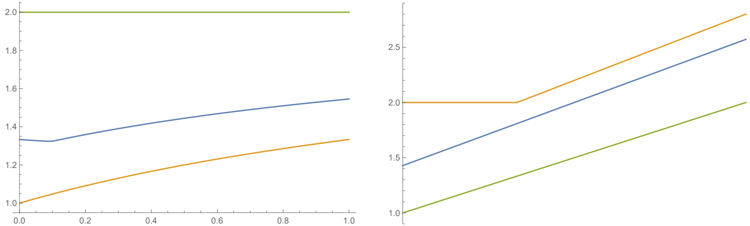


Figure: Left: Worst-case ratios of LPT and theoretic lower bounds for $1|nr-pm|C_{max}$. From top to bottom: bounds given in Ji, He, and T. Cheng, 2007, bounds given in this paper, and the theoretic lower bound.
Right: Worst-case ratios of algorithms and theoretic lower bounds for $P2|nr-pm|C_{max}$. From top to bottom: bounds given in Sun and H. Li, 2010, bounds given in this paper, and the theoretic lower bound.

Lemma

For any instance of $Pm|nr-pm|C_{max}$, we can construct a companion one-dimensional bin packing instance by making the following substitutions.

- Instance of scheduling problem $\Longleftrightarrow$ Instance of bin-packing problem
- Job $\Longleftrightarrow$ Item
- Processing time of a job $\Longleftrightarrow$ Size of an item
- Availability period $\Longleftrightarrow$ Bin
- Length of each availability period $\Longleftrightarrow$ Capacity of each bin

Lemma

Bin-packing Problem

Given a set of n items with sizes $p_1, p_2, \cdots, p_n$, all items should be packed into a number of bins and the sum of the sizes of items being packed into each bin is at most T.

Algorithm First Fit Decreasing(FFD for short)

Reorder all items such that $p_1 \geq p_2 \geq \cdots \geq p_n$, then packed each item into the first bin it fits.

Lemma

- b^{FFD} (b for short) and b^*_{BP}: the number of bins created by *FFD* and in an optimal packing, respectively.

- B_i: the ith bin created by the *FFD* algorithm, $i = 1, \ldots, b$ / the set of jobs that are packed in it.

- b^*: the number of availability periods that at least one job is processed on in the optimal schedule.

Lemma

Lemma 4

(Dósa, 2007) $b \leq \frac{11}{9} b^*_{BP} + \frac{6}{9}$ *and the bound is tight.*

Lemma

Let $\mathcal{B} = \{B_1, B_2, \ldots, B_{b-1}\}$, and $\mathcal{B}_I$, $\mathcal{B}_{II}^1$, $\mathcal{B}_{II}^2$, $\mathcal{B}_{III}$ be disjoint subsets of $\mathcal{B}$. Concretely,
(i) $\mathcal{B}_I$ consists of bins of $\mathcal{B}$ which contains exactly one item.
(ii) $\mathcal{B}_{II}^1$ consists of bins of $\mathcal{B}$ which contains exactly two items with one of them has a size greater than $\frac{T}{2}$.
(iii)$\mathcal{B}_{II}^2$ consists of bins of $\mathcal{B}$ which contains exactly two items with none of them has a size greater than $\frac{T}{2}$.
(iv) $\mathcal{B}_{III}$ consists of bins of $\mathcal{B}$ which contains exactly three items.
If $p_n > \frac{T}{4}$, then any bin can contain at most 3 items. Thus $\mathcal{B} = \mathcal{B}_I \cup \mathcal{B}_{II}^1 \cup \mathcal{B}_{II}^2 \cup \mathcal{B}_{III}$.
Let y_0 be the number of items packed to B_b.

Lamma

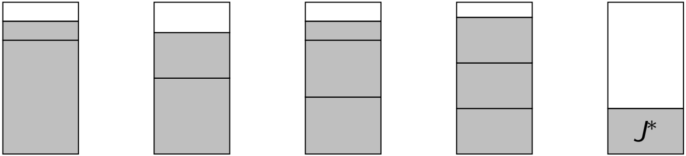
Lemma 5

Suppose that $b > b^*_{BP} \geq 2$ *and* J_n *is packed in* B_b *with* $p_n > \frac{T}{4}$.
(i) If $b - b^*_{BP} = 2k + 1$, *where* k *is an integer* ($b - b^*_{BP}$ *is an odd number), then* $|\mathcal{B}^2_{II}| \geq 6k + y_0$ *and* $|\mathcal{B}_{III}| \geq 8k + y_0$.
(ii) If $b - b^*_{BP} = 2k + 2$, *where* k *is an integer*($b - b^*_{BP}$ *is an even number), then* $|\mathcal{B}^2_{II}| \geq 6k + 3 + y_0$ *and* $|\mathcal{B}_{III}| \geq 8k + 4 + y_0$.

What dose lemma 5 mean

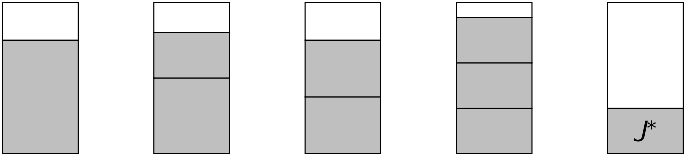
Let J^* denote the smallest item in B_b, p^* denote its size. For any instance I, if J^* is not the smallest item in $\mathcal{J}$,
we can obtain instance II by deleting items which are smaller than J^*.

FFD(I)

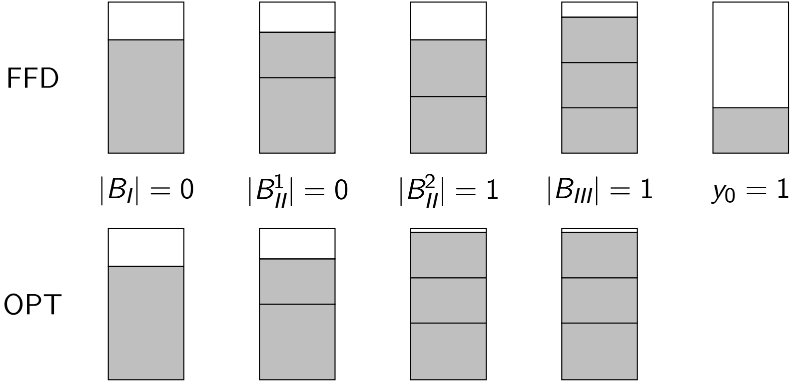


$C^*(I) \geq C^*(II)$ when considering the corresponding scheduling instanc

FFD(II)



What dose lemma 5 mean



An illustration of the case where $b = 3$ and $b^*_{BP} = 2$.

Why we need Lemma 5

To get a good enough lower bound for C^* !!!

Let $|B_i|$ denote the total processing time of jobs in B_i and P denote the total processing time of $\mathcal{J}$.

• $p^* \leq \frac{T}{4}$: Then $|B_i| > \frac{3}{4}T$ for i=1,2,...b-1. Therefore, $P > (b-1)\frac{3}{4}T + |B_b|$ and we can get an estimate of C^* by this argument.

• $p^* > \frac{T}{4}$: If we prove each bin have to contain at least three jobs in the optimal solution, we would have a fairly satisfatory lower bound. Hence, we consider the number of jobs of an instance where b and b^*_{BP} are fixed.

Why we need Lemma 5

Clearly, with lemma 5, we can enumberate all possible triples $(|\mathcal{B}_{II}^2|, |\mathcal{B}_{III}|, y_0)$. Then we get the number of different kinds of jobs and we can estimate C^* by this imformation. For example, still $b = 3$ and $b_{BP}^* = 2$, and we know $(|\mathcal{B}_{II}^2|, |\mathcal{B}_{III}|, y_0) = (1, 1, 1)$ according to lemma 5. Therefore, there are at least three jobs in each bin in optimal solution and $C^* > \frac{3}{4}T$.

Proof Outline of Lemma 5

- $\mathcal{J}_I = \{J_i | p_i > T - p^*, J_i \in \mathcal{J}\}$;
- $\mathcal{J}_{II}^1 = \{J_i | p_i > \frac{T}{2}, J_i \in \mathcal{J}\}$;
- $\mathcal{J}_{II}^2$ denote a set consisting of items contained by bins in $\mathcal{B}_{II}^2$;
- $\mathcal{J}_{III}$ denote a set consisting of items contained by bins in $\mathcal{B}_{III}$.

Proof Outline of Lemma 5

In the optimal solution, it is easy to show that there are $|\mathcal{B}_I|$ bins contain items in $\mathcal{J}_I$ and there are $|\mathcal{B}_{II}^1|$ bins contain items in $|\mathcal{J}_{II}^1|$. Other bins can contain at most three items as $p_n > \frac{T}{4}$, which implies that:

$$n = |\mathcal{B}_I| + 2(|\mathcal{B}_{II}^1| + |\mathcal{B}_{II}^2|) + 3|\mathcal{B}_{III}| + y_0 \leq |\mathcal{B}_I| + 2|\mathcal{B}_{II}^1| + 3(b_{BP}^* - |\mathcal{B}_I| - |\mathcal{B}_{II}^1|) \quad (1)$$

As $b = |\mathcal{B}_I| + |\mathcal{B}_{II}^1| + |\mathcal{B}_{II}^2| + |\mathcal{B}_{III}| + 1$,

$$b - b_{BP}^* \leq 1 + \frac{|\mathcal{B}_{II}^2| - y_0}{3}. \quad (2)$$

Then we prove

$$|\mathcal{B}_{II}^2| \geq 6k + y_0 / 6k + 3 + y_0$$

Proof Outline of Lemma 5

- The total size of any two items in J_{II}^2 is greater than $T - p^*$.

- There is an optimal solution σ^* satisfying: For any bin B_i in $B_I \cup B_{II}^1$, there is a bin B_i^* containing same items as B_i. Without loss of generality, we consider σ^* only.

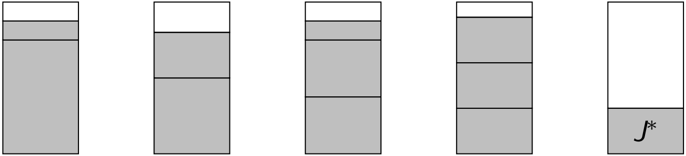
- In order to packed items into less bins, some bins contain only one item in J_{II}^2 as each bin contain at most two items if both of them are in J_{II}^2. (Otherwise, we need $|B_{II}^2|$ bins to contain J_{II}^2, $|B_{III}|$ bins to contain J_{III}. Therefore, $b = b^*$.)

- By dicussing the number of bins containing only items in B_{III}, we get the lower bound of $|B_{III}|$.

$1|nr-pm|C_{max}$

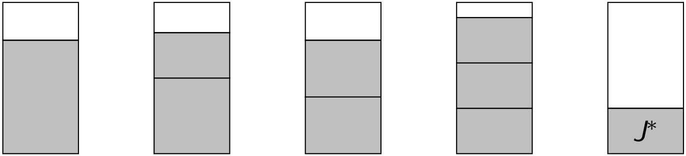
Since deleting jobs which are smaller than p^* would not change the value of C^A, we only discuss instances where $p_1 \geq p_2 \ldots \geq p_n = p^*$.

FFD(I)



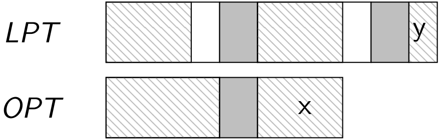
$C^*(I) \geq C^*(II)$ when considering the corresponding scheduling instance

FFD(II)



$1|nr-pm|C_{max}$

Let y denote the sum of processing time of jobs in the last batch in the solution obtained by the LPT algorithm and x denote the sum of processing time of jobs in the last batch in the solution obtained by the optimal algorithm.



$1|nr-pm|C_{max}$

The upper bound of the worst-case ratio of Algorithm LPT:

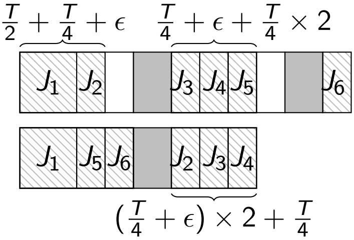
$$r(\beta)=\begin{cases}\frac{44+44\beta}{33+36\beta}, & \beta\in(0,\frac{\sqrt{313}-15}{32}]\approx(0,0.0841],\\ \frac{29+28\beta}{22+20\beta}, & \beta\in(\frac{\sqrt{313}-15}{32},\frac{\sqrt{181}-11}{24}]\approx(0.0841,0.1022],\\ \frac{9+8\beta}{7+4\beta}, & \beta\in(\frac{\sqrt{181}-11}{24},\infty)\approx(0.1022,\infty),\end{cases}$$

$1|nr-pm|C_{max}$

The tightness when $\beta \geq 0.1022$ can be proved by the following instance. Let $\mathcal{J} = \{J_1, J_2, J_3, J_4, J_5, J_6\}$, $p_1 = \frac{T}{2}$, $p_2 = p_3 = \frac{T}{4} + \epsilon$, $p_4 = p_5 = p_6 = \frac{T}{4}$, $\epsilon > 0$. Then, $C^A = \frac{9}{4}T + 2t$, $C^* = \frac{7}{4}T + 2\epsilon + t$.

$C^A = \frac{9}{4}T + 2t$

$C^* = \frac{7}{4}T + 2\epsilon + t$



$1|nr-pm|C_{max}$

Note that $b^*_{BP}=b^*$ in this case. Here b^* is the number of availability periods that at least one job is processed on in the optimal schedule.

- If $b=b^*\geq 2$,

$y-x<y-(P-(b^*-1)T)<y-(y+(b-1)(T-p^*)-(b^*-1)T)$
$=(b^*-1)p^*$.

$$\frac{C^A}{C^*}=\frac{(b^*-1)(T+t)+y}{(b^*-1)(T+t)+x}=1+\frac{y-x}{(b^*-1)(T+t)+x}$$
$$<1+\frac{(b^*-1)p_n}{(b^*-1)(T+t)+p_n}$$

By discussing the case of $p^*\leq\frac{T}{b^*}$ and $p^*>\frac{T}{b^*}$, we prove $\frac{C^A}{C^*}\leq\frac{4+2\beta}{3+2\beta}$.

$1|nr-pm|C_{max}$

- If $b > b^*$:
  - If $p^* \leq \frac{T}{4}$

$$\begin{aligned}\frac{C^A}{C^*} &= \frac{(b-1)(T+t)+y}{(b^*-1)(T+t)+x} \\ &< \frac{(b-1)(T+t)+y}{(b^*-1)(T+t)+(b-1)(T-p_n)-(b^*-1)T+y} \\ &= \frac{(b-1)(T+t)+y}{(b^*-1)t+(b-1)(T-p_n)+y}.\end{aligned} \quad (3)$$

Combining $b \leq \lfloor \frac{11}{9} b^*_{BP} + \frac{6}{9} \rfloor$, we obtain that $\frac{C^A}{C^*} \leq g(b^*)$, where

$$g(b^*) = \frac{(\frac{11}{9}b^* - \frac{3}{9})(T+t) + \frac{T}{4}}{(b^*-1)t + (\frac{11}{9}b^* - \frac{3}{9})\frac{3}{4}T + \frac{T}{4}}$$

$1|nr-pm|C_{max}$

- If $p^* > \frac{T}{4}$:

  - If $b^* \geq 11$ and $b^* \neq 14$, $b < \lfloor \frac{11}{9}b^* + \frac{6}{9} \rfloor$, otherwise $b \geq |B_{II}^2| + |B_{III}| + 1 > b$ according to lemma 5.

  - If $b^* \geq 11$ and $b < \lfloor \frac{11}{9}b^* + \frac{6}{9} \rfloor$, then

$$\frac{C^A}{C^*} \leq \frac{(\lfloor \frac{11}{9}b^* + \frac{6}{9} \rfloor - 2)(T+t) + y}{(b^*-1)(T+t) + x} < \frac{((\frac{11}{9}b^* + \frac{6}{9}) - 2)(T+t) + T}{(b^*-1)(T+t) + \frac{T}{4}}$$

$$\leq r(\beta)$$

$1|nr - pm|C_{max}$

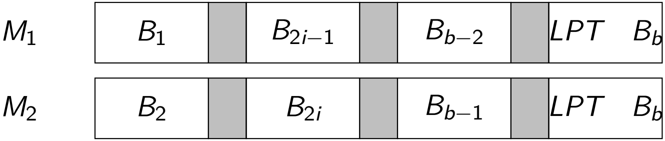
- For the remaining cases, the main technique of this part is enumberating all values of $(|B_{II}^2|, |B_{III}|, y_0)$ according to lemma 5 and estimating C^A and C^*, which is similar to the analysis we have done for $b^* = 2$, $b = 3$ before. Detailed description of this part of the proof will take a lot of time, so it will not be explained here.

$P2|nr - pm|C_{max}$

Algorithm DFFD

1. Apply the *FFD* algorithm for the companion bin-packing instance. If $b = 2k + 1$, where k is an integer, Go to Step 2. If $b = 2k$, where k is an integer, Go to Step 3.
2. For $i = 1, \ldots, k$, process the jobs in B_{2i-1} on the ith availability period of M_1, and processing jobs in B_{2i} on the ith availability period of M_2. Process the jobs in B_b on two machines by *LPT* algorithm. Output the resulting schedule. Stop.

$P2|nr-pm|C_{max}$



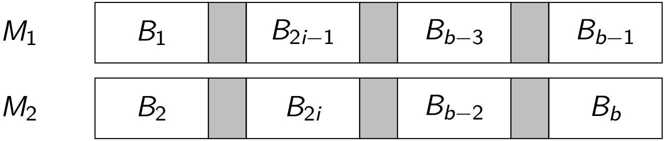
An illustration of the case of $b \equiv 1 \pmod 2$, where B_i denotes the set of jobs placed in the ith bin.

$P2|nr-pm|C_{max}$

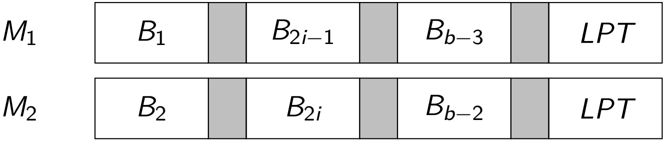
Algorithm DFFD

3. For $i = 1, \ldots, k$, process the jobs in B_{2i-1} on the ith availability period of M_1, and processing jobs in B_{2i} on the ith availability period of M_2. Denote the resulting schedule by σ_1.
4. For $i = 1, \ldots, k-1$, process the jobs in B_{2i-1} on the ith availability period of M_1, and processing jobs in B_{2i} on the ith availability period of M_2. Process the jobs in $B_{b-1} \cup B_b$ on two machines by LPT algorithm. Denote the resulting schedule by σ_2.
5. Select the better schedule of σ_1 and σ_2 as output. Stop.

$P2|nr - pm|C_{max}$



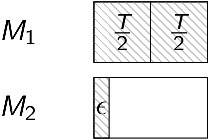
An illustration of the σ_1, where $b \equiv 2 \pmod 2$.



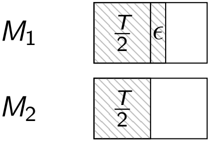
An illustration of the σ_2, where $b \equiv 2 \pmod 2$.

$P2|nr-pm|C_{max}$

It's necessary to obtain both σ_1 and σ_2. Let's consider the instance with $p_1 = p_2 = \frac{T}{2}$ and $p_3 = \epsilon$.



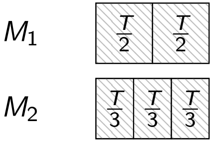
If we only consider σ_1, $\frac{C^A}{C^*} = \frac{T}{\frac{T}{2}+\epsilon} \rightarrow 2$.



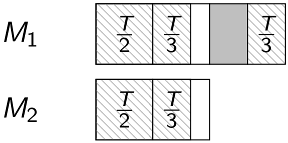
If we only consider σ_2, $C^A = C^*$

$P2|nr-pm|C_{max}$

There are also instances where σ_1 is better than σ_2



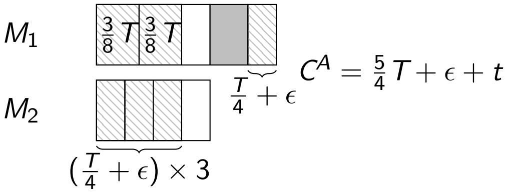
If we only consider σ_1, $C^A = C^*$.

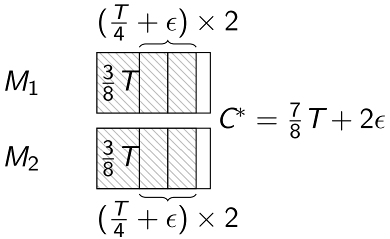


If we only consider σ_2, $\frac{C^A}{C^*} = \frac{\frac{7}{6}T+t}{T}$

$P2|nr-pm|C_{max}$

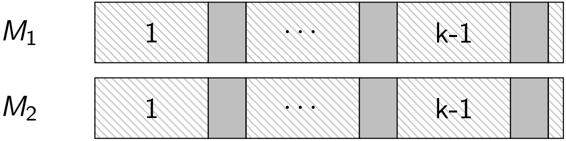
The worst case ratio of DFFD is $\frac{10}{7}+\frac{8}{7}\beta$. The tightness of it can be proved by the following instance.



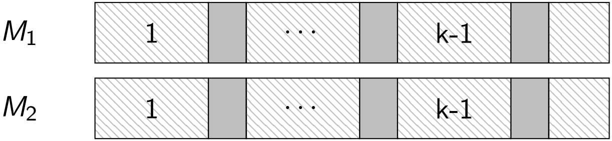


$P2|nr - pm|C_{max}$

In this model, b^* may not equal to b^*_{BP}. Besides, we redefine b^* in order to establish a clearer connection between P and b^*.



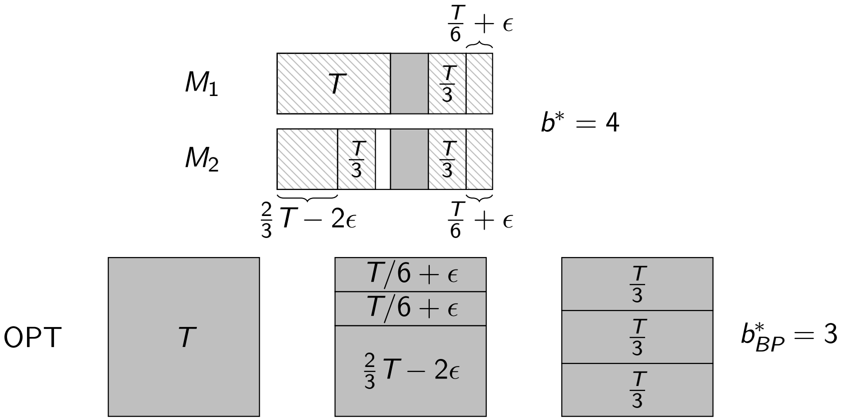
$$b^* = 2k - 1$$



$$b^* = 2k$$

$P2|nr - pm|C_{max}$

Note that, even if b^* have been redefined, it still may not equal to b^*_{BP}.



$P2|nr-pm|C_{max}$

Lemma 6

$$\frac{C^A}{C^*} \leq \frac{(b+1)T+(b-1)t}{(b^*-1)T+(b^*-2)t}$$

$P2|nr-pm|C_{max}$

Combining lemma 6 and $b \leq \lfloor \frac{11}{9}b^* + \frac{6}{9} \rfloor$, we prove if $b^* \geq 13$, $\frac{C^A}{C^*} \leq \frac{10}{7} + \frac{8}{7}\beta$.

The proof of remaining cases is tedious, since we enumberate all possible value of (b, b^*) and discuss each case respectively. Therefore, we only show main techniques used in this part.

$P2|nr-pm|C_{max}$

Lemma 7

If items corresponding to $J_1, \cdots, J_k$ are packed in two bins by the FFD algorithm, $p_1 \geq p_2 \cdots \geq p_k$, $k \geq 3$ and $P = p_1 + \cdots + p_k \leq 2T - p_3$, then $J_1, \cdots, J_k$ are processed in one batch on each machine if we apply the LPT algorithm.

Lemma 8

If $b = 2k + 1$ and the processing time of jobs in the last batch on both machines is no more than p_0, then
(i) if $y_0 \leq 2$, then $C^A \leq kT + p_0$.
(ii) if $y_0 \geq 3$, then $C^A \leq kT + \frac{T+p_0}{2}$.
More generally, if jobs in $\mathcal{J}$ are processed on M_1 and M_2 by the LPT algorithm and the processing time of the last finished job is no more than p_0. Then $C^A \leq \frac{P+p_0}{2}$ if $P \leq 2T - p_3$.

$P2|nr - pm|C_{max}$

- If $b \equiv 1 \pmod 2$ or $b \equiv 0 \pmod 2$ and P is small enough, we analyse σ_2 and use lemma 8 to get the upper bound of C^A.

- Otherwise, we analyse σ_1 and take T as the upper bound of the total processing time of jobs in the last available period.

$P2|nr-pm|C_{max}$

For estimation of C^*:

• We take $\frac{P}{2}$ or $k^*T+\frac{T}{2}$ as the lower bound of C^*.

• If technique above dose not work, we discuss the case of $p^* \leq \frac{T}{4}$ and the case of $p^* > \frac{T}{4}$.

• For the case of $p^* > \frac{T}{4}$, we deleting jobs which are smaller than p^* to get a new instance and use lemma 5 to get the lower bound of C^*.

• For the case of $p^* \leq \frac{T}{4}$, we obtain a lower bound of P and then a lower bound of C^*.

$P2|nr - pm|C_{max}$

The last theorem is about the theoretic lower bound of $P2|nr - pm|C_{max}$.

Theorem 9

For $P2|nr - pm|C_{max}$, there is no polynomial time approximation algorithm with a worst-case ratio of less than $1 + \beta$ unless $P = NP$.

PARTITION: Given n positive integers $h_1, h_2, \cdots, h_n$ with $\sum_{i=1}^{n} h_i = 2H$, does there exist a set $U \subseteq \{1, 2, \cdots, n\}$, with $\sum_{i \in U} h_i = H$?

We prove that there is no polynomial time approximation algorithm with a worst-case ratio of less than $1 + \beta$ by showing that if not, then the algorithm can be used to establish a polynomial time algorithm for solving the PARTITION problem, which is NP-hard.

Conclusion

In conclusion, we give an upper bound of the worst-case ratio of the LPT algorithm with β as parameter to $1|nr-pm|C_{max}$. What's more, we propose a new algorithm, Algorithm DFFD, which beats the existing algorithm to $P2|nr-pm|C_{max}$. And we give the upper bound of the worst case ratio of Algorithm DFFD which is tight. Finally, we give a theoretic lower bound for $P2|nr-pm|C_{max}$.

Reference I

Dósa, György (2007). "The tight bound of first fit decreasing bin-packing algorithm is FFD(I) 11/9OPT(I) + 6/9". In: *Combinatorics, Algorithms, Probabilistic and Experimental Methodologies, Lectuer Notes in Computer Science*. 4614, pp. 1–11.

Gawiejnowicz, Stanisław (2020a). "A review of four decades of time-dependent scheduling: main results, new topics, and open problems". In: *Journal of Scheduling* 23, pp. 3–47.

– (2020b). *Models and Algorithms of Time-Dependent Scheduling*. Springer, Berlin, Heidelberg.

Ji, Min, Yong He, and T.C.Edwin Cheng (2007). "Single-machine scheduling with periodic maintenance to minimize makespan". In: *Computers & Operations Research* 34.6, pp. 1764–1770.

Reference II

Lee, Chung-Yee (2004). "Machine scheduling with an availability constraint". In: *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Ed. by Joseph Y-T Leung. CRC.

Li, Ganggang and Xiwen Lu (2015). "Two-machine scheduling with periodic availability constraints to minimize makespan". In: *Journal of Industrial and Management Optimization* 11.2, pp. 685–700.

Ma, Ying, Chengbin Chu, and Chunrong Zuo (2010). "A survey of scheduling with deterministic machine availability constraints". In: *Computers & Industrial Engineering* 58.2, pp. 199–211.

Qi, Xiangtong (2007). "A note on worst-case performance of heuristics for maintenance scheduling problems". In: *Discrete Applied Mathematics* 155.3, pp. 416–422.

Reference III

Qi, Xiangtong, Tsiushuang Chen, and Fengsheng Tu (1999). "Scheduling the maintenance on a single machine". In: *Journal of the Operational Research Society* 50.10, pp. 1071–1078.

Sun, Kaibiao and Hongxing Li (2010). "Scheduling problems with multiple maintenance activities and non-preemptive jobs on two identical parallel machines". In: *International Journal of Production Economics* 124.1, pp. 151–158.

Xu, Dehua, Zhenmin Cheng, et al. (2009). "Makespan minimization for two parallel machines scheduling with a periodic availability constraint". In: *Computers & Operations Research* 36, pp. 1809–1812.

Xu, Dehua, Kaibiao Sun, and Hongxing Li (2008). "Parallel machine scheduling with almost periodic maintenance and non-preemptive jobs to minimize makespan". In: *Computers & Operations Research* 35.4, pp. 1344–1349.

Reference IV

Xu, Dehua, Yunqiang Yin, and Hongxing Li (2009). "A note on 'scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan'". In: *European Journal of Operational Research* 197.2, pp. 825–827.

Yu, Xianyu, Yulin Zhang, and George Steiner (2014). "Single-machine scheduling with periodic maintenance to minimize makespan revisited". In: *Journal of Scheduling* 17.3, pp. 263–270.

Thank You!