



Scheduling Many Types of Calibrations

Hua Chen¹, Vincent Chau^{2(✉)}, Lin Chen³, and Guochuan Zhang¹

¹ Zhejiang University, Hangzhou, China
{chenhua_by, zgc}@zju.edu.cn

² Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences,
Shenzhen, China
vincentchau@siat.ac.cn

³ Texas Tech University, Lubbock, TX, USA
chenlin198662@gmail.com

Abstract. Machines usually require maintenance after a fixed period. We need to perform a calibration before using the machine again. Such an operation requires a non-negligible cost. Thus finding a schedule minimizing the total cost of calibrations is of great importance.

This paper studies the following scheduling problem. We have a single machine, n jobs where each job j is characterized by its release time r_j , deadline d_j , and processing time p_j . Moreover, there are K types of calibrations, i.e., when the machine performs a calibration of type $k \in \{1, \dots, K\}$ instantaneously, it can maintain calibrated for a fixed length T_k with a corresponding cost f_k . Jobs can only be processed when the machine is in the calibrated state. Our goal is to find a feasible schedule that minimizes the total cost of calibrations.

We consider two classes of models: the costs of the calibrations are arbitrary, and the costs of the calibrations are equal to their length. For the first model, we propose a pseudo-polynomial time algorithm and a $(2 + \epsilon)$ -approximation algorithm when jobs have agreeable deadlines (later release time implies a later deadline). For the second model, we give a 2-approximation algorithm.

Keywords: Scheduling · Calibration · Approximation algorithms

1 Introduction

Scheduling is one of the most classical and important problems in combinatorial optimization. Recently a class of scheduling problems related to calibrations has been brought up by Bender et al. in their seminal paper [3]. The motivation of the problem comes from the Integrated Stockpile Evaluation (ISE) problem [4]. ISE is a program to test nuclear weapons so that they can function normally. Operating these tests needs precision, or safety mistakes can produce a significant

Hua Chen and Guochuan Zhang are supported by NSFC (No. 11531014). Vincent Chau is supported by the CAS President's International Fellowship Initiative n° 2020FYT0002, 2018PT0004.

loss. Meanwhile, there are testing machines for testing weapons. The testing machines need to be calibrated after running a fixed period to ensure that the testing tasks are processed smoothly.

Similarly, the calibration scheduling problem can be seen as a multi-agents game. For example, during a game, the agents may have abrasion resulting in the inaccurate shooting and need to be calibrated after using for a while. Generally, every agent can decide to charge after some time, and every charging has its corresponding cost and working time.

We formally define the ISE problem as follows: we are given a set J of n jobs (weapons) and m identical machines (testing machines). Each job $j \in J$ is defined by its release time r_j , its deadline d_j and its processing time p_j . We calibrate a machine instantaneously, and the machine can stay valid for $T \geq 2$ time units. The scheduling of all jobs must be feasible, i.e., (1) each job must be scheduled on one of the m identical machines and must be scheduled during p_j calibrated slots, (2) each job must be entirely scheduled between its release time r_j and its deadline d_j and (3) one machine can only process one job at the same time. The goal is to find a feasible schedule using the minimum number of calibrations, where a feasible schedule requires that the scheduling of all jobs should be feasible, and the calibrations used are non-overlapping.

Using the 3-field notation developed in [10], the problem can be denoted as $P|r_j, d_j, p_j, T|\#(\text{calibrations})$.

Related Work

Bender et al. [3] studied the problem in which jobs have unit processing time. They gave a polynomial time algorithm to compute the optimal solution, while a 2-approximation algorithm is given for the multiple machine case. They pointed out that the complexity of the problem remained unknown. Recently, Chen et al. [8] proved that when the number of machines m is constant, the problem can be solved polynomially with dynamic programming. On the other hand, when m is part of the input, they gave a PTAS (polynomial-time approximation scheme).

Later, Fineman and Sheridan [9] considered the case in which jobs have arbitrary processing time, and the preemption of jobs is not allowed¹. Note that it is NP-hard to decide whether a feasible schedule exists since it can be reduced from the decision version of the bin packing problem. They considered a resource-augmentation version of the problem, and they related it to the classical machine minimization problem [13]. When preemption of jobs is allowed, Angel et al. [1] generalized the algorithm from [3] and showed that it could be solved in polynomial time.

Chau et al. [5] considered the flow time problem with calibrations. They focused on the online version whose objective is to minimize the total flow time, the elapsed time between the release time of a job until its completion, as well as the calibration cost. They aimed to find a tradeoff between the flow time and the cost of the calibrations, and they gave several constant competitive online algorithms for different settings. Wang [14] studied the time-slot cost

¹ A job is not allowed to be interrupted once it has been started.

variant of the scheduling problem with calibrations. The cost of scheduling a job depends on the starting time. The goal is to compute a schedule of minimum cost with at most B calibrations. Wang [14] proposed dynamic programmings for different scenarios of this variant. Chau et al. [6] investigated the throughput variant of this scheduling problem: the goal is to maximize the total profit of scheduled jobs. They showed that the problem admits a constant approximation algorithm for arbitrary processing time jobs. Finally, Chau et al. [7] considered that calibrations could only occur simultaneously. They showed that the problem could be solved in polynomial time by giving a dynamic programming algorithm. They also proposed some fast approximation algorithms depending on the cost function of a batch of calibrations.

All the above problems considered one type of calibration. When there are K types of calibrations with respective length T_k and respective cost f_k for $k \in \{1, \dots, K\}$, Angel et al. [1] proved that when jobs have unit processing time, the problem can be solved in polynomial time by providing a dynamic programming algorithm. However, when jobs have arbitrary processing time, the problem becomes NP-hard. They showed for the particular case in which all the jobs have the same release time and the same deadline. This particular case is similar to the Knapsack Cover Problem for which there exists a $(1 + \varepsilon)$ -approximation algorithm [11].

Scheduling with calibrations has similarities with some other well-known scheduling problems, such as minimizing idle periods [2], and scheduling on cloud-based machines which must be rented to perform work [12].

Our Contributions

In this paper, we study the scheduling problem with K types of calibrations on a single machine. We have n jobs where each job j is characterized by its release time r_j , deadline d_j , and processing time p_j . Moreover, there are K types of calibrations, i.e., when the machine performs a calibration of type $k \in \{1, \dots, K\}$ instantaneously, it can maintain calibrated for a fixed length T_k with a corresponding cost f_k . Jobs can only be processed when the machine is in the calibrated state. Our goal is to find a feasible schedule that minimizes the total cost of calibrations.

The problem is NP-hard even if all the jobs have common release time and common deadline. We investigate the following two generalized cases:

- *arbitrary calibration cost*: the cost of the calibrations does not depend on its length. In this work, we assume that every single job can entirely be scheduled into a single calibration, i.e., $\max_j p_j \leq \min_k T_k$, and jobs have *agreeable deadline*, i.e., for every pair of jobs i, j , we have $r_i \leq r_j$, if and only if $d_i \leq d_j$. In particular, we establish:
 - a pseudo-polynomial time algorithm whose running time is $O(n^5 K^2 P^2 f_{\min}^2)$ where $P = \sum_{j=1}^n p_j$ and $f_{\min} = \min_k f_k$ in Sect. 2.1.
 - a $(2 + \varepsilon)$ -approximation algorithm in Sect. 2.2.
- *uniform calibration cost*: the cost of the calibrations is equal to its length. For this case, we give a 2-approximation algorithm in Sect. 3.

In the sequel, we suppose without loss of generality that jobs are sorted in non-decreasing order of their deadlines, i.e., $d_1 \leq d_2 \leq \dots \leq d_n$. Similarly, we sort the calibration types in the non-decreasing order of their length, i.e., $T_1 < T_2 < \dots < T_K$. Without loss of generality, we also have $f_1 < f_2 < \dots < f_K$.

2 Arbitrary Calibration Cost

In this section, we investigate the problem with arbitrary calibration cost by proposing dynamic programming algorithms. We first give a pseudo-polynomial time algorithm, then we show how to adapt it into a polynomial running time by losing a constant factor on the objective function.

2.1 A Pseudo-Polynomial Time Algorithm

We first define several time points that are pertinent in any schedule. In [1], they showed some properties for the unit processing time jobs case. We obtain the following by dividing the jobs into unit processing time jobs, i.e., for each job j , we replace by p_j jobs with unit processing time.

Let $\Phi := \{d_j - h \mid j = 1, \dots, n; h = 1, \dots, P\}$.

Proposition 1 (Proposition 1 [1]). *There exists an optimal schedule in which calibrations start at a time in Φ .*

In the sequel, we only consider schedules satisfying Proposition 1. Moreover, since jobs have agreeable deadlines, we have the following proposition.

Proposition 2 (Lemma 1 [14]). *There exists an optimal solution in which jobs are scheduled in the non-decreasing order of their deadline.*

Let $\mathcal{F} := \{f_{\min}, f_{\min} + 1, \dots, nf_{\min}\}$ be the set of cost of any schedule where $f_{\min} = \min_k f_k$.

Because each job can fit into a single calibration, we know that the cost of the optimal solution OPT is at least $f_{\min}$ and at most $nf_{\min}$, so $OPT \in \mathcal{F}$. We are now ready to describe our dynamic programming.

Dynamic Programming. Let $c(j, f, t, k)$ be the minimum completion time of job j in a feasible schedule whose cost is at most f , such that:

- the first j jobs ($\{1, \dots, j\}$) are scheduled into the opened calibrations;
- the starting time of the last calibration is t ;
- the type of the last calibration is k .

The idea of our dynamic programming is to compute the number of available time slots in the last calibration. Because jobs have agreeable deadline, we know that the job j will be scheduled after the job $j - 1$. The idea is to schedule the job j as early as possible in order to get the minimum completion time. We distinguish three cases (See Fig. 1 for an illustration of the different cases):

1. Job j is scheduled in the same calibration as the completion time of job $j - 1$.
 - (a) The job j starts immediately after the completion of the job $j - 1$. The job j will be scheduled from $c(j - 1, f, t, k)$ to $c(j - 1, f, t, k) + p_j$.
 - (b) There are some idle time slots between the completion time of the job $j - 1$ and the release time of the job j . Then, the job j is scheduled from r_j to $r_j + p_j$.
2. Job j is scheduled in two calibrations: it starts in the same calibration as the completion of job $j - 1$, and ends in another calibration.
 - (a) The job j starts immediately after the completion of the job $j - 1$. The job j will be executed from $c(j - 1, f', t', k')$ to $t' + T_{k'}$, then from t to $t + p_j - (t' + T_{k'} - c(j - 1, f', t', k'))$.
 - (b) There are some idle time slots between the completion time of the job $j - 1$ and the release time of the job j . Then, the job j starts from r_j to the end of the calibration at $t' + T_{k'}$, then from t to $t + p_j - (t' + T_{k'} - r_j)$.
3. Job j is scheduled in a different calibration as job $j - 1$.
 - (a) The job j starts in a different calibration containing the job $j - 1$ and it starts at time t . The job j is executed from t to $t + p_j$.
 - (b) The job j is executed in a different calibration containing the job $j - 1$ and starts at its release time $r_j > t$. Thus, the job j is executed from r_j to $r_j + p_j$.

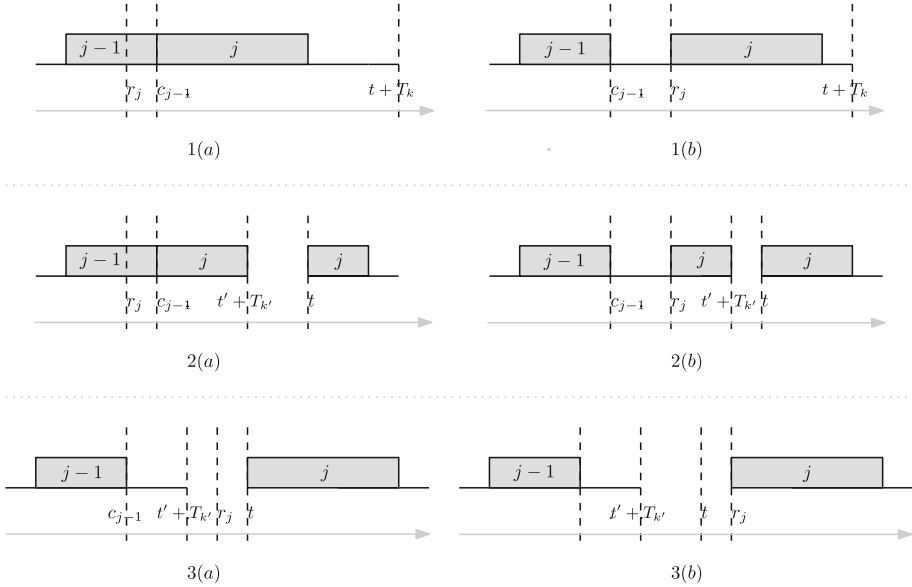


Fig. 1. Illustration of different cases for scheduling job j in the dynamic programming. c_{j-1} denotes the completion time of job $j - 1$. In 1(a) and 1(b), $c_{j-1} := c(j - 1, f', t, k)$. In the remaining cases, we have $c_{j-1} := c(j - 1, f', t', k')$.

Hence, we have the following recursive function.

Proposition 3. *By convention, if the schedule is not feasible, the completion time of such a schedule is $+\infty$. We have $c(j, f, t, k)$*

$$= \min \begin{cases} \min\{c_j \mid c_j = p_j + \max\{c(j-1, f, t, k), r_j\}, c_j \leq \min\{d_j, t + T_k\}\} \\ \min \left\{ c_j \mid \begin{array}{l} c_j = p_j - t' - T_{k'} + \max\{c(j-1, f', t', k'), r_j\} + t, c_j \leq d_j, \\ r_j < t' + T_{k'}, f' + f_k \leq f, f' \in \mathcal{F}, t' \in \Phi, k' \in \{1, \dots, K\} \end{array} \right\} \\ \min\{c_j \mid c_j = \max\{t, r_j\} + p_j, c_j \leq d_j, r_j \geq t' + T_{k'}, f' + f_k \leq f\} \\ +\infty \end{cases}$$

We initialize the table as follows:

$$c(1, f, t, k) = \min \begin{cases} \max\{t, r_1\} + p_1, & \text{if } f \geq f_k \text{ and } \max\{t, r_1\} + p_1 \leq d_1, \\ +\infty, & \text{otherwise.} \end{cases}$$

The objective is to find the minimum cost f^* such that $c(n, f^*, t, k) \leq d_n$ for $f^* \in \mathcal{F}, t \in \Phi, k \in \{1, \dots, K\}$.

Theorem 1. *The dynamic programming algorithm in Proposition 3 computes an optimal solution for the arbitrary calibration cost scheduling problem.*

Proof. If the jobs $\{1, \dots, j\}$ cannot be scheduled into the opened calibrations whose total cost is at most f , then the schedule is not feasible, and we have $c(j, f, t, k) = +\infty$. In particular, if there are not enough time slots for the job j , i.e., $c(j, f, t, k) > d_j$, then the of the schedule is $+\infty$. It corresponds to the last line of the dynamic program.

We prove the claim by showing that in the dynamic program, we have tried every possibility of scheduling the job j , as well as the starting time of the calibrations. As described previously, we have six forms in mathematics (We assume $c(j, f, t, k) \leq d_j$ in the following):

1. Job j is scheduled in the same calibration as the completion of job $j-1$.
 - (a) If $r_j < c(j-1, f, t, k)$, then $c(j, f, t, k) = c(j-1, f', t, k) + p_j$.
 - (b) If $r_j \geq c(j-1, f, t, k)$, then $c(j, f, t, k) = r_j + p_j$.
2. Job j is scheduled into two calibrations. Since we need to open a new calibration, we need to ensure that the schedule of the first $j-1$ jobs is of cost at most $f' = f - f_k$.
 - (a) If $r_j < t' + T_{k'}$ and $c(j-1, f', t', k') > r_j$, it means that the job j is scheduled right after the completion time of job $j-1$ until the end of the current calibration, and then the remaining part of the job j , which is equal to $p_j - (t' + T_{k'} - c(j-1, f', t', k'))$, is scheduled in the last calibration that starts at time t . Hence, $c(j, f, t, k) = \min_{f', t', k'} \{p_j - (t' + T_{k'} - c(j-1, f', t', k')) + t\}$.

- (b) If $r_j < t' + T_{k'}$ and $c(j-1, f', t', k') \leq r_j$, which means that the remaining part of the job, which is equal to $p_j - (t' + T_{k'} - r_j)$, is scheduled in a new calibration of type k starting at time t . Hence, $c(j, f, t, k) = \min_{f', t', k'} \{p_j - (t' + T_{k'} - r_j) + t\}$.
3. Job j starts in a different calibration as job $j-1$. As the previous case, we need to open a new calibration, and we need to ensure that the schedule of the first $j-1$ jobs is of cost at most $f' = f - f_k$.
- (a) If $r_j \geq t' + T_{k'}$ and $t > r_j$, it means that we have to schedule the job j no earlier than t , and thus $c(j, f, t, k) = t + p_j$.
- (b) If $r_j \geq t' + T_{k'}$ and $t \leq r_j$, which means that the job j starts no earlier than its release time r_j , thus $c(j, f, t, k) = r_j + p_j$.

The optimal value is $\min \left\{ f \mid \left\{ c(n, f, t, k) : f \in \mathcal{F}, t \in \Phi, k \in \{1, \dots, K\} \right\} \right\}$.

According to the dynamic programming algorithm above, its running time is $O(nK^2|\Phi|^2|\mathcal{F}|^2) = O(n^5K^2P^2f_{\min}^2)$, due to $|\Phi| = O(nP)$ and $|\mathcal{F}| = O(nf_{\min})$.

Hence, the running time is pseudo-polynomial. $\square$

2.2 A Constant Approximation Algorithm

To achieve a polynomial time algorithm, we aim to avoid going through all different parameter values in the dynamic program. So, we focus on the sets $\mathcal{F}$ and Φ whose sizes are pseudo-polynomial. We aim to reduce the size of such sets. We define the set of different objective values of the schedules as $\mathcal{F}'$.

Let $\mathcal{F}' := \{f_{\min} \cdot (1 + \varepsilon)^q \mid q = 0, \dots, \lceil \log_{1+\varepsilon} n \rceil\}$. We have $|\mathcal{F}'| = O(\log_{1+\varepsilon} n)$. We now show that considering the values in $\mathcal{F}'$ can lead to a solution whose cost is no more than $(1 + \varepsilon)$ times of optimal cost OPT .

Lemma 1. *If we restrict the cost f to $\mathcal{F}'$ and assume f^* attains the minimum value in all schedules of $\{c(n, f, t, k) \mid f \in \mathcal{F}', t \in \Phi, k \in \{1, \dots, K\}\}$ after using the dynamic programming for $f \in \mathcal{F}', t \in \Phi, k \in \{1, \dots, K\}$, then we have $f^* \leq (1 + \varepsilon)OPT$.*

Proof. We know that $f_{\min} \leq OPT \leq nf_{\min}$. Then there exists a q_0 such that $f_{\min} \cdot (1 + \varepsilon)^{q_0-1} \leq OPT \leq f_{\min} \cdot (1 + \varepsilon)^{q_0}$. Thus, $OPT \leq f_{\min} \cdot (1 + \varepsilon)^{q_0} \leq OPT \cdot (1 + \varepsilon)$. Since $f^* \leq f_{\min} \cdot (1 + \varepsilon)^{q_0}$, we obtain $f^* \leq (1 + \varepsilon)OPT$. $\square$

Similarly, we define the new set of the starting times of the calibrations as $\Phi' := \{d_j - aT_1 \mid j = 1, \dots, n; a = 0, \dots, n\}$.

Note that we have $|\Phi'| = O(n^2)$. Next, we show that if we restrict the starting times of calibrations to Φ' and do not restrict the costs, then a solution with no more than twice the optimal cost exists. We initially allow to have overlapping calibrations (a time slot can be covered by more than one calibration), which will be handled later without increasing the solution's cost.

Lemma 2. *For the scheduling problem with arbitrary calibration cost, there exists a 2-approximate solution such that the calibrations start at a time in Φ' .*

Proof. We denote OPT to be the optimal value of the problem. Let O be an optimal solution verifying Proposition 1, and we denote the sequence of calibrations in O as $\{C_{i_1}^O, C_{i_2}^O, \dots\}$, where $C_{i_u}^O$ represents the u -th calibration in O and its type is i_u where $i_u \in \{1, \dots, K\}$.

We show that when we restrict the starting times of the calibrations in S to Φ' , we can get our desired conclusion. Let $C_{i_u}^O$ be a calibration in O such that it does not start at a time in Φ' . We replace it by two calibrations of the same type such that the first one starts at a time in Φ' and such that they cover (at least) the same interval as initially.

Since the schedule O verifies Proposition 1, it means that the starting time of the calibration $C_{i_u}^O$ is at most at a distance of P from a deadline d_h . We assume the distance between d_h and the starting time of $C_{i_u}^O$ is ℓ . See Fig. 2.

Moreover, we have $\min_j p_j \leq T_1$, so we have $\ell \leq P \leq nT_1$, so there is an integer point $t = d_h - aT_1 \in [d_h - \ell - T_{i_u}, d_h - \ell] \cap \Phi'$, where $a \in \{0, \dots, n\}$. Thus, we can replace such calibration by two calibrations starting respectively at t and $t + T_{i_u}$. See Fig. 3.

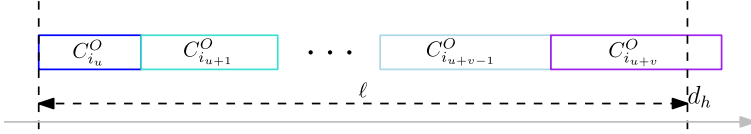


Fig. 2. Illustration of the calibration $C_{i_u}^O$ in O .

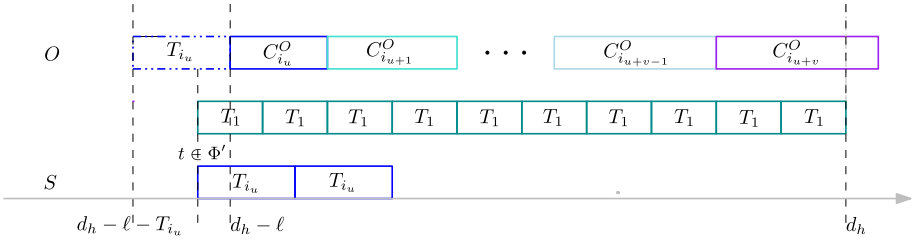


Fig. 3. Illustration of relation of O and the constructed S .

We repeat such modification as long as there is a calibration that does not start at a time in Φ' in the schedule (except the newly added calibrations).

For every calibration in O , there are two consecutive identical calibrations whose types are the same, and the starting time of the first one is in Φ' . Then, all the jobs stay at the same time as in O . Hence, the cost of such schedule is $2OPT$. $\square$

Note that the schedule is feasible for jobs but not for the calibrations since there may exist calibrations that overlap each other. To make all calibrations used be non-overlapping, we need the following observation.

Observation 1. *We can transform a schedule with overlapping calibrations into a schedule without overlapping calibrations in polynomial time without increasing the cost of the solution.*

Indeed, when two calibrations overlap, we can change the starting time of the one that starts later to start when the first calibration ends. Meanwhile, all the jobs stay scheduled at their initial time. We modify at most $2n$ calibrations. Figure 4 illustrates an example to handle two overlapping calibrations.

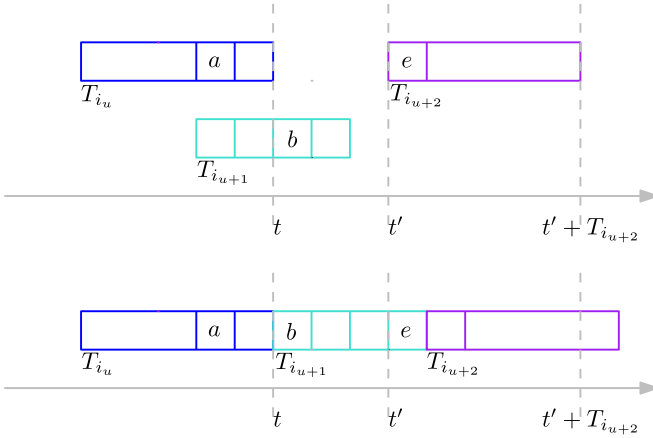


Fig. 4. An example of handling the overlapping calibrations.

The running time of the dynamic program is pseudo-polynomial in the number of choices of cost. Because of Lemma 2, we need to redefine the range of $\mathcal{F}'$ to $\mathcal{F}''$, where $\mathcal{F}'' := \{2f_{\min} \cdot (1 + \varepsilon)^q \mid q = 0, \dots, \lceil \log_{1+\varepsilon} n \rceil\}$.

First, we discretize the choices of the cost to $\mathcal{F}''$ instead of $\mathcal{F}'$. Then, we force the starting times of calibrations to be in a set Φ' . Finally, we use two calibrations every time to ensure consistency with the constructed S in Lemma 2.

Now we will show that when we restrict the cost to $\mathcal{F}''$ and the starting times of calibrations to Φ' , there exists a feasible solution whose cost is no more than $(2 + \varepsilon)OPT$.

Modified Dynamic Programming (MDP). We modify the dynamic programming proposed in Proposition 3 as follows:

- we restrict the choices of the cost to $\mathcal{F}''$;
- we restrict the choices of the starting times of calibrations to Φ' ;

- when a new job comes, there are two possibilities:
 - there is not any new calibration to increase;
 - increase two new consecutive calibrations whose types are identical.

The objective is to find the smallest f^* such that $c(n, f, t, k) \leq d_n$ for $f \in \mathcal{F}'', t \in \Phi', k \in \{1, \dots, K\}$.

Theorem 2. *For the problem with arbitrary calibration cost, MDP computes a feasible schedule whose cost is no more than $(2 + \varepsilon)$ times of the optimal value, for $\varepsilon > 0$ arbitrarily small. The running time of MDP is polynomial in n and in $1/\varepsilon$.*

Proof. In order to prove the theorem, we combine Lemma 1 with Lemma 2. Note that the schedule returned by MDP allows the calibrations to overlap. So the cost f^* is a lower bound of the cost of such a schedule. Moreover, we have $f^* \leq 2OPT \cdot (1 + \varepsilon)$ by Lemma 1. So we have $f^* \leq 2(1 + \varepsilon)OPT = (2 + \varepsilon')OPT$, where $\varepsilon' = 2\varepsilon$.

Finally, we perform the same operations according to Observation 1 to make all calibrations non-overlapping. Thus, we get a feasible schedule of cost no more than f^* , which completes the proof of the approximation ratio $(2 + \varepsilon)$ for $\varepsilon > 0$ arbitrarily small.

Running Time. Since $|\mathcal{F}''| = O(\log_{1+\varepsilon} n)$, $|\Phi'| = O(n^2)$, and $|\{1, \dots, K\}| = K$, the size of the table of MDP is $n \cdot |\mathcal{F}''| \cdot |\Phi'| \cdot |\{1, \dots, K\}| = O(n^3 K \log_{1+\varepsilon} n)$. When the values of the table are fixed, the minimization is over the values f', t' and k' , so the running time is $O(n^2 K \log_{1+\varepsilon} n)$. Hence, the overall time complexity is $O(n^5 K^2 \log_{1+\varepsilon}^2 n)$. $\square$

3 Uniform Calibration Cost

In this section, we consider the case in which the calibration cost is equal to its length. We show that there exists a 2-approximation algorithm. In particular, we use the Preemptive Lazy Binning (PLB) algorithm [1] with the shortest calibration.

Theorem 3. *For the problem $1|r_j, d_j, pmtn, \{T_1, \dots, T_K\} | cost(calibrations)$ with $f_k = T_k$ for all $k \in \{1, \dots, K\}$, PLB algorithm with the shortest calibration is a 2-approximate.*

Proof. Suppose we have the optimal sequence of calibrations type $O = \{T_{i_1}, T_{i_2}, \dots\}$. Then we construct a feasible solution in which we only use the type 1.

We replace all the calibrations in O with type 1 such that they cover the initial calibrations $\{T_{i_1}, T_{i_2}, \dots\}$, i.e., if a calibration has length T_k , then we replace it with $\lceil T_k/T_1 \rceil$ calibrations of type 1. See S' in Fig. 5. Jobs stay scheduled at the same time. Similarly, if some calibrations overlap, then we perform the operations as in Observation 1. A non-overlapping schedule S shown in Fig. 5 is obtained.

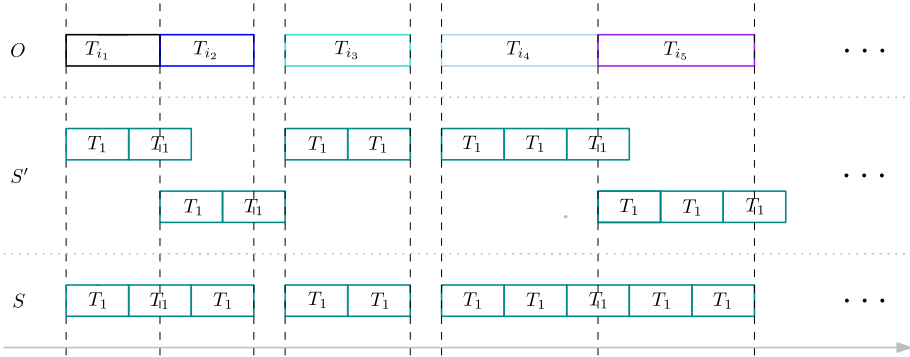


Fig. 5. Illustration of analysis.

Note that for any T_{i_u} in O , the new constructed schedule S' above will cover it by at most $T_{i_u} + T_1 < 2T_k$ since $T_1 < T_k$. So the cost of the non-overlapping schedule S is no more than the cost of S' whose cost is less than twice of the optimal value.

We recall that the *PLB* algorithm proposed in [1] returns a schedule with the minimum number of calibrations in polynomial time when there is only one type of calibration. If we are only allowed to use the shortest calibration of length T_1 , S is a feasible schedule, while *PLB* algorithm returns a schedule with at most as many calibrations as in S .

Hence, by using the *PLB* algorithm with the shortest calibration, the cost of the returned schedule is no more than the cost of S and less than twice of the optimal value. $\square$

4 Conclusion

In this paper, we studied the scheduling problem with multiple types of calibration. When jobs have agreeable deadlines, we showed that the problem could be solved in pseudo-polynomial time, and we gave a $(2 + \varepsilon)$ -approximation algorithm. We further studied the case in which the calibration cost is equal to its length, and we gave a simple 2-approximation algorithm. A natural question is to improve the approximation ratio. It would be mostly interesting if one can show a PTAS or exclude its existence.

Moreover, no approximation algorithm is known for the general case of jobs, a constant approximation algorithm would be of great interest.

References

1. Angel, E., Bampis, E., Chau, V., Zissimopoulos, V.: On the complexity of minimizing the total calibration cost. In: Xiao, M., Rosamond, F. (eds.) FAW 2017. LNCS, vol. 10336, pp. 1–12. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59605-1_1
2. Baptiste, P.: Scheduling unit tasks to minimize the number of idle periods: a polynomial time algorithm for offline dynamic power management. In: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA), pp. 364–367. Society for Industrial and Applied Mathematics (2006)
3. Bender, M.A., Bunde, D.P., Leung, V.J., McCauley, S., Phillips, C.A.: Efficient scheduling to minimize calibrations. In: Proceedings of the 25th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), pp. 280–287 (2013)
4. Burroughs, C.: New integrated stockpile evaluation program to better ensure weapons stockpile safety, security, reliability (2006). <http://www.sandia.gov/LabNews/060331.html>
5. Chau, V., Li, M., McCauley, S., Wang, K.: Minimizing total weighted flow time with calibrations. In: Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), pp. 67–76. ACM (2017)
6. Chau, V., Feng, S., Li, M., Wang, Y., Zhang, G., Zhang, Y.: Weighted throughput maximization with calibrations. In: Friggstad, Z., Sack, J.-R., Salavatipour, M.R. (eds.) WADS 2019. LNCS, vol. 11646, pp. 311–324. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24766-9_23
7. Chau, V., Li, M., Wang, E.Y., Zhang, R., Zhao, Y.: Minimizing the cost of batch calibrations. *Theor. Comput. Sci.* **828–829**, 55–64 (2020)
8. Chen, L., Li, M., Lin, G., Wang, K.: Brief announcement: approximation of scheduling with calibrations on multiple machines. In: Proceedings of the 31st ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), pp. 237–239. ACM (2019)
9. Fineman, J.T., Sheridan, B.: Scheduling non-unit jobs to minimize calibrations. In: Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), pp. 161–170 (2015)
10. Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.R.: Optimization and approximation in deterministic sequencing and scheduling: a survey. In: *Annals of Discrete Mathematics*, vol. 5, pp. 287–326. Elsevier (1979)
11. Ibarra, O.H., Kim, C.E.: Fast approximation algorithms for the Knapsack and sum of subset problems. *J. ACM* **22**(4), 463–468 (1975)
12. Mäcker, A., Malatyali, M., Meyer auf der Heide, F., Riechers, S.: Cost-efficient scheduling on machines from the cloud. *J. Comb. Optimiz.* **36**(4), 1168–1194 (2017). <https://doi.org/10.1007/s10878-017-0198-x>
13. Phillips, C.A., Stein, C., Torng, E., Wein, J.: Optimal time-critical scheduling via resource augmentation. In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC), pp. 140–149 (1997)
14. Wang, K.: Calibration scheduling with time slot cost. *Theor. Comput. Sci.* **821**, 1–14 (2020)