Single machine scheduling: finding the Pareto Set for jobs with equal processing times with respect to criteria $L_{\text{max}}$ and $C_{\text{max}}$.

Alexander Lazarev · Dmitry Arkhipov · Frank Werner

1 Introduction

In this paper, the special case of the classical NP-hard scheduling problem $1|r_j|L_{\text{max}}$ is considered. There is a single machine and a set of jobs $N = \{1, 2, \ldots, n\}$ to be executed with identical processing times $p_j = p$ for all jobs $j \in N$. We define a schedule (or sequence) $\pi$ as the execution sequence $K_1(\pi), K_2(\pi), \ldots, K_n(\pi)$, where

$$K_1(\pi) \cup K_2(\pi) \cup \cdots \cup K_n(\pi) \equiv N.$$ 

The equality $K_i(\pi) = j$ means that job $j \in N$ is executed under the ordinal number $i$ in the schedule $\pi$. The execution of the job $K_i(\pi) = j$ starts at time

$$R_j(\pi) = \max\{C_{K_i-1}(\pi), r_{K_i}(\pi)\}$$

(where $C_{K_0}(\pi) = 0$) and finishes at time

$$R_j(\pi) + p = C_j(\pi),$$

where $C_j(\pi)$ is the completion time of the job $j \in N$. Let us denote the lateness of job $j$ under the schedule $\pi$ as

$$L_j(\pi) = C_j(\pi) - d_j.$$ 

The maximum completion time and the maximum lateness are denoted as $C_{\text{max}}$ and $L_{\text{max}}$, respectively. Let us call the schedule $\pi$ allowable for the set $N$ if all jobs according to the schedule $\pi$ execute without preemptions and intersections. We denote the set of all
allowable schedules as $\Pi$. The goal is to find a feasible schedule $\pi \in \Pi$, which satisfies the following optimization criterion:

$$\min_{\pi \in \Pi} \max_{j \in N} L_j(\pi).$$

2 The auxiliary problem

Let us formulate an auxiliary problem. We consider the same set of jobs $N = \{1, 2, \ldots, n\}$ and a bound on the maximum lateness $y$. The goal is to construct a schedule satisfying the following optimization criterion:

$$\min_{\pi \in \Pi} \max_{j \in N} C_j(\pi)| L_{\text{max}}(\pi) < y. $$

For each set of due dates $d_1, \ldots, d_n$ and the bound on the lateness $y$, deadlines $D_j$ can be calculated by the following formula:

$$D_j = d_j + y.$$

An allowable schedule satisfying this restriction is called feasible. To construct the solution of the auxiliary problem, we consider the approach presented in [3]. Next, we briefly recall the main idea from this paper.

The auxiliary algorithm works as follows. While the completion times of all jobs are lower than its deadlines, schedule the jobs according to the algorithm, presented in [4]. If for any job $X \in N$, the inequality

$$C_X \geq D_X$$

holds, then execute the special procedure $\text{CRISIS}(X)$. This procedure finds the job $A$, which is already scheduled with the latest completion time, but for which

$$D_A > D_X$$

holds. This job is called $\text{Pull}(X)$ and all jobs which are already scheduled after $\text{Pull}(X)$ and $X$ constitute the restricted set $S(A, X)$. We define $r_{S(A, X)}$ to be the earliest time when the jobs of $S(A, X)$ can start their execution. The procedure $\text{CRISIS}(X)$ reschedules the jobs of the set $\{A\} \cup S(A, X)$. The procedure fails when a job $\text{Pull}(X)$ for a crisis job $X$ does not exist. After a successful execution of the procedure $\text{CRISIS}(X)$, Schrage’s algorithm [4] is used to schedule the jobs. Such a scheduling is repeated until any call of the procedure $\text{CRISIS}()$ fails or all jobs from the set $N$ have been successfully scheduled.

3 Solution of the main problem

Next, we consider the main problem $1\mid r_j, p_j = p\mid L_{\text{max}}$. We also present an algorithm to obtain the Pareto set of schedules with respect to the criteria $L_{\text{max}}$ and $C_{\text{max}}$. First, we introduce a procedure $\text{CHECK}(\pi, N, y)$ which constructs the schedule $\pi^*$ as follows.

$$\text{CHECK}(\pi, N, y)$$

1. Set the lateness bound $y$ and a time $t = \min_{i \in N} r_i$.
2. Set the deadlines $D_i := d_i + y$.
3. If all jobs from the set $N$ have been scheduled, go to step 7.
4. While $t$ is not in the interval $[r_{S(A, X)}, D_X]$ for any restricted set $S(A, X)$ from the schedule $\pi$ that has not yet been completely performed, execute the jobs under $\pi^*$ according to Schrage’s algorithm.
5. Otherwise, execute only the jobs from the set $S(A, X)$ under the schedule $\pi$, and then go to step 3.
6. If in steps 4-5 any job $Y$ experiences a crisis, run the procedure $\text{CRISIS}(Y)$.

7. return ($\pi^*$).

**Lemma 1** Let $\pi$ and $\pi'$ be the schedules constructed by the auxiliary algorithm for the bounds $y$ and $y'$, respectively, and

$$\pi^* = \text{CHECK}(\pi, N, y).$$

If $y < y'$, then

$$\pi^* = \pi'.$$

holds.

Next, we describe the main algorithm $M$ to obtain the Pareto set with respect to the criteria $L_{\text{max}}$ and $C_{\text{max}}$.

**MAIN ALGORITHM** (Algorithm $M$)

1. Set the bound $y_1 := +\infty$.
2. Construct the schedule $\pi_1$ according to the auxiliary algorithm, and add it to $\Phi$, i.e.: $\Phi := \{\pi_1\}$; set the counter $k := 1$; set the bound $y_1 := L_{\text{max}}(\pi_1)$.
3. Construct the schedule $\pi_{k+1} = \text{CHECK}(\pi_k, N, y_k)$.
   a) If the schedule $\text{CHECK}(\pi_k, N, y)$ exists, then:
      add $\pi_{k+1}$ to the set $\Phi$, i.e.: $\Phi := \Phi \cup \pi_k$; set $y_k = L_{\text{max}}(\pi_k)$;
      increase the counter $k$, i.e.: $k := k + 1$; repeat step 3.
   b) Otherwise, return ($\Phi$).

At last, we formulate and prove some important lemmas and a theorem, which show that algorithm $M$ finds the Pareto set $\Phi$ in $O(n^2 \log n)$ operations.

**Lemma 2** If any job becomes a crisis job for the second time, then the algorithm stops.

**Theorem 1** After the execution of Algorithm $M$, the Pareto set of schedules $\Phi$ according to the criteria $L_{\text{max}}$ and $C_{\text{max}}$ has been constructed, where the schedules $\Phi_1$ and $\Phi_{|\Phi|}$ are optimal according to criteria $L_{\text{max}}$ and $C_{\text{max}}$ respectively. For this set

$$|\Phi| \leq n + 1$$

holds.

**Lemma 3** The complexity of Algorithm $M$ is $O(n^2 \log n)$.

**4 Metric analysis**

The metric $\rho$ for the instances of problem $1|r_j|L_{\text{max}}$ was introduced in [5]. We estimate a metric distance $\rho^p(A)$ between an arbitrary instance $A$ which holds $p_1^A \leq \cdots \leq p_n^A$, and a set of polynomial solvable instances with the identical processing times of jobs as:

$$\rho^p(A) \leq \sum_{i=1}^{\lfloor(n-1)/2\rfloor} p_{n-i+1}^A - p_i^A.$$

The prove that estimated bound is tight and present a polynomial algorithm to find the instance $B$ for an arbitrary instance $A$ which satisfy

$$\rho(A, B) = \rho^p(A).$$

The results of numerical experiments are also presented.
References


