MISTA 2015

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

Alexander Lazarev · Dmitry Arkhipov · Frank Werner

1 Introduction

In this paper, the special case of the classical NP-hard scheduling problem $1|r_j|L_{max}$ 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) π 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 π. 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 π as

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

The maximum completion time and the maximum lateness are denoted as $C_{\max}$ and $L_{\max}$, respectively. Let us call the schedule π *allowable* for the set N if all jobs according to the schedule π execute without preemptions and intersections. We denote the set of all

Alexander Lazarev
V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russian Federation;
Lomonosov Moscow State University, Moscow, Russian Federation;
Moscow Institute of Physics and Technology, Dolgoprudny,
Russian Federation;
National Research University Higher School of Economics,Moscow, Russian Federation
E-mail: jobmath@mail.ru

Dmitry Arkhipov
V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russian Federation
E-mail: miptrafter@gmail.com

Frank Werner
Faculty of Mathematics, Institute of Mathematical Optimization, Otto-von-Guericke University Magdeburg, Germany
E-mail: frank.werner@ovgu.de

allowable schedules as Π. 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:

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

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 $CRISIS(X)$. 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 $Pull(X)$ and all jobs which are already scheduled after $Pull(X)$ 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 $CRISIS(X)$ reschedules the jobs of the set $\{A\} \cup S(A, X]$. The procedure fails when a job $Pull(X)$ for a crisis job X does not exist. After a successful execution of the procedure $CRISIS(X)$, Schrage's algorithm [4] is used to schedule the jobs. Such a scheduling is repeated until any call of the procedure $CRISIS()$ fails or all jobs from the set N have been successfully scheduled.

3 Solution of the main problem

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

$CHECK(\pi, N, y)$

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 π that has not yet been completely performed, execute the jobs under π^* according to Schrage's algorithm.
5. Otherwise, execute only the jobs from the set $S(A, X]$ under the schedule π, and then go to step 3.

6. If in steps 4-5 any job Y experiences a crisis, run the procedure $CRISIS(Y)$.
7. $return(\pi^*)$.

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

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

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_{max} and C_{max}.

MAIN ALGORITHM (Algorithm M)

1. Set the bound $y_0 := +\infty$.
2. Construct the schedule π_1 according to the auxiliary algorithm,
   and add it to Φ, i.e.: $\Phi := \{\pi_1\}$;
   set the counter $k := 1$;
   set the bound $y_1 := L_{max}(\pi_1)$.
3. Construct the schedule $\pi_{k+1} = CHECK(\pi_k, N, y_k)$.
   a) If the schedule $CHECK(\pi_k, N, y)$ exists, then:
      add π_{k+1} to the set Φ, i.e.: $\Phi := \Phi \cup \pi_k$;
      set $y_k = L_{max}(\pi_k)$;
      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 Φ 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 Φ according to the criteria L_{max} and C_{max} has been constructed, where the schedules Φ_1 and $\Phi_{|\Phi|}$ are optimal according to criteria L_{max} and C_{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 ρ for the instances of problem $1|r_j|L_{max}$ 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:

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

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

1. Kravchenko, S.A. and Werner, F.: Parallel Machine Problems with Equal Processing Times. *Journal of Scheduling*. Vol. 14, No. 5, 2011, 435 - 444.
2. Garey, M.R.; Johnson, D.S.; Simons, B.B. and Tarjan, R.E.: Scheduling unit-time tasks with arbitrary release times and deadlines. *SIAM J. COMPUT*. Vol. 10, No. 2, May 1981, 256 - 269.
3. Simons, B.B.: A fast algorithm for single processor scheduling. *In 19th Annual Symposium on Foundations of Computer Science (Ann Arbor, Mich., 1978)*, 246-252.
4. Schrage, L.: Solving Resource-Constrained Network Problems by Implicit Enumeration: Non Preemptive Case. *Operations Research*. Vol. 18 Issue 2, 1970, 263 - 278.
5. Lazarev, A.A.: The Pareto-optimal set of the NP-hard problem of minimization of the maximum lateness for a single machine. *Journal of Computer and Systems Sciences International. M.: SP MAIK Nauka/Interperiodica*. 2006. 45, No. 6. . 943-949.