Polynomial algorithm for the scheduling problem $1|pmtn, p = 2, r_j = j - 1, w_j \le w_{j+1}| \sum w_j c_j$

A.A. Lazarev, D.I Arkhipov
Institute of control science, Moscow
Russian foundation for basic research
jobmath@mail.ru, miptrafter@gmail.com

December 5, 2011

Abstract

In this paper, we consider the following scheduling problem. On the single machine need to process n jobs. Job $j, j = 1, \dots, n$, characterized: release times $r_j = j - 1$; processing time $p_j = 2$; weight of jobs are non-decreasing $w_j \le w_{j+1}$.
The objective function is $\min \sum_{j=1}^{n} (w_j \cdot C_j)$, where C_i – completion time.

Statement 1. *The machine does not idle without work.*
This statement follows from the fact that in each moment there is a job which is available to execute.

Statement 2. *When the job j starts to realize there exists the job i which also started it's realization, and hasn't completed yet. Then job j should be completed before the job i continues it's execution.*

Statement 3. *All jobs are able to start it's execution only in integer times.*

Corollary 1. *a)According to 3 and 1 all jobs must start and finish only in integer times.*
b)All jobs must process only in integer time intervals (1 or 2)

Statement 4. *If the job does not get started in it release time, it would start after r_n.*

Lemma 1. *Assume the job i starts it's execution at moment r_i, and $c_i \le r_n$. Then in interval $[r_i + 1, c_i - 1]$ all jobs is executed without interruption.*

Definition 1. *We define for each job i the variable $a_i = r_{\mu(i)}$ - receive time of the job $\mu(i)$ which satisfies the following inequality $w_{\mu(i)-1} \le 2w_i \le w_\mu$. If such job $\mu(i)$ doesn't exists we define $a_i = n$.*

Lemma 2. *The job i started it's execution at moment r_i,then was interrupted and completed at time $c_i > r_n$. Then, all jobs received in the interval $[r_i+1; a_i)$ can not execute in the interval $[n + 1; c_i)$*

Lemma 3. *Job i starts it's realization at the moment r_i, then was interrupted and completed at time $c_i > r_n$. Then there are no jobs which started it's execution in the interval $[r_i + 1; a_i)$,then were interrupted and competed it's execution before the moment a_i*

Corollary 2. *If the job i starts it realization at the moment r_i, then was interrupted and completed at time $c_i > r_n$. Then we have not more than one job which started in it's receive time r_j ($r_i < r_j < a_i$) and then was interrupted.*
This job should be completed before the $n + 1$.

Algorithm 1. *Now, let us show you the algorithm for constructing an optimal schedule.*
We define $\pi(i)$ as the optimal schedule for the jobs with weights $\{w_i, w_{i+1} \dots w_n\}$ and release dates $\{r_i, r_{i+1} \dots r_n\}$. Note that our goal is to construct the schedule $\pi(1)$.
We will construct schedules $\pi(n), \pi(n-1) \dots \pi(1)$ step by step.

Step 1
$\pi(n) = n; n$
Step n + 1 - i
We know the schedules $\pi(i + 1), \dots \pi(n - 1), \pi(n)$. Let us show how to construct the schedule πi.
Let's consider some cases:
1)Both units of the job i complete instantly.
According to the Smith's rule and 4 we obtain that the optimal schedule is $\{i, i, \pi(i + 2), i + 1, i + 1\}$.
2a)The job i starts it realization at the moment r_i, then was interrupted and continued its realization at the moment $t \le r_n$.
According to 1 all jobs in the interval (r_i+1, t) couldn't be interrupted. Then, due to the Smith's rule and the 4 all jobs which were received but haven't

started before the moment t must start their execution when all jobs with receive times $\geq t+1$ will be completed.
We obtain that the optimal schedule for each t is: $\{i, i+1, i+1, i+3, i+3 \ldots t-1, t-1, i, \pi(t+2), t+1, t+1, t, t, \ldots i+2, i+2\}$
2b)The job i starts it realization at the moment r_i, then was interrupted and continued its realization at the moment $t > r_n$.
2b.1)There are no interruptions in the interval $(r_i+1; a_i)$. The job $\mu(i)$ starts its execution at the moment moment $a_i = r_{\mu(i)}$.
Note that each job j such as $i < j < \mu(i)$ could either be completed at the moment a_i or start its execution after the moment c_i. Hence, in the interval $(a_i; c_i - 1)$ to execute only jobs $\{a_i + 1, \ldots n\}$. Hence, the optimal schedule for this case is:
$\{i, i+1, i+1, i+3, i+3 \ldots a_i - 1, a_i - 1, \pi(\mu(i)), i, a_i, a_i, \ldots i+2, i+2\}$
2b.2)There are no interruptions in the interval $(r_i+1; a_i)$. The job $\mu(i)$ starts its execution at the moment moment $a_i = r_{\mu(i)} + 1$
We have the one difference between this case and 2b.1) related to parity of the interval $(r_i; r_{\mu(i)})$. So, the optimal schedule is:
$\{i, i+1, i+1, i+3, i+3 \ldots a_i, a_i, \pi(\mu(i)+1), \mu(i), \mu(i), i, a_i, a_i, \ldots i+2, i+2\}$
2b.3)There is an interruption in the interval $(r_i + 1, a_i)$.
According to the 2 & 2 that there are can be only one interrupted job $(r_i + 1, a_i)$ which must be completed before the moment $n + 1$. Let this job be performed beginning at t and continued its execution at $t' < n$. Note that there are no interruptions in the intervals $(r_i + 1; t)\&(t; t')$(due to the 1). According to the Smith's rule, 2 and the fact that $t' \geq a_i$ we have the optimal schedule for each pair of jobs $\{t, t'\}$:
$\{i, i+1, i+1 \ldots t-1, t-1, t+1, t+2, t+2, t'-1, t'-1, t+1, \pi(t'+2)$, *all jobs with the receive dates from the interval $(a_i, t'+2)$ (which haven't started yet), i, all jobs with receive dates lower than a_i (which haven't started yet)*$\}$

We obtained on each step:
1 schedule in case 1)
About n schedules (for each t) in case 2a)
1 schedule in case 2b.1) and 1 schedule in 2b.2)
About n^2 schedules schedules in 2b.3) (for each pair $\{t, t'\}$). So, we obtain about $O(n^2)$ schedules on each step and we also need $O(n)$ operations to sort the jobs according to Smith's rule.
After $n-1$ steps we obtain the schedule $\pi(1)$.
So, the complexity of this algorithm equals $O(n^4)$.

References

[1] P. Baptiste Scheduling Equal-Length Jobs on Identical Parallel Machines. Discrete Appl. Math., Number 103, 2000, 21–32.

[2] A.A. Lazarev, A.G. Kvaratskhelia Properties of Optimal Schedules for the Minimization Total Weighted Completion Time in Preemptive Equal-length Job with Release Dates Scheduling Problem on a Single Machine// Automation and Remote Control, Vol. 71, Number 10, 2010, 2085–2092.