Approximation algorithms for single machine scheduling with non-renewable resources and the total weighted completion time*

Péter Györgyi (Speaker) † Tamás Kis ‡

1 Introduction

We consider single machine scheduling problems with additional non-renewable resource constraints. Non-renewable resources (like raw materials, energy or money) are consumed by the jobs when they are started, while there is an initial stock, and some additional supplies in the future with known supply dates and quantities. A number of papers examined the problem under various objective functions, however, there are hardly any approximability results with the total weighted completion time objective, which is the topic of this paper.

Formally, we have a single machine, a set of n jobs $\mathcal{J}$, and a non-renewable resource. Each job j has a processing time $p_j > 0$, a weight $w_j > 0$, and a resource requirement $a_j \geq 0$. The initial stock from the resource is $b_1 \geq 0$ at time $u_1 = 0$, and there are replenishments at dates $0 < u_2 < \ldots < u_q$ in quantities $b_\ell \geq 0$ for $\ell = 2, \ldots, q$. We can assume that the total demand equals to the total supply ($\sum_{j \in \mathcal{J}} a_j = \sum_{\ell=1}^{q} b_\ell$). A schedule specifies the starting time S_j of each job $j \in \mathcal{J}$; it is feasible if (i) no pair of jobs overlap in time ($S_{j_1} + p_{j_1} \leq S_{j_2}$ or $S_{j_2} + p_{j_2} \leq S_{j_1}$ for each $j_1 \neq j_2 \in \mathcal{J}$), and (ii) for each time point t, the total supply until time t is not less than the total consumption of those jobs starting not later than t, i.e., if $u_\ell \leq t$ is the last supply date before t, then $\sum_{j \in J : S_j \leq t} a_j \leq \sum_{\ell'=1}^{\ell} b_{\ell'}$. We aim at finding a feasible schedule S that minimizes the total weighted completion time $\sum_{j \in \mathcal{J}} w_j C_j$. We denote our problem using the standard $\alpha|\beta|\gamma$ notation by $1|nr = 1|\sum w_j C_j$, where '$nr = 1$' indicates that we have only one type of non-renewable resource.

For a given sequence of jobs, one can easily determine a schedule in which there is idle time before a job j only if there is not enough resource left on stock to start

*Supported by the GINOP-2.3.2-15-2016-00002 grant of the Ministry of National Economy of Hungary.

†gyorgyi.peter@sztaki.mta.hu. Institute for Computer Science and Control, Kende str. 13-17, H-1111 Budapest, Hungary

‡kis.tamas@sztaki.mta.hu. Institute for Computer Science and Control, Kende str. 13-17, H-1111 Budapest, Hungary

j after finishing the last job before the idle period, and thus it has to wait for new supplies.

The above problem is interesting both from the practical as well as from the theoretical point of view. Consider for instance the preparation of the weekly schedule of a production line, where some of the raw materials built into the products arrive over the week, and the supplies constrain what and when can be produced. On the other hand, the problem has a strong connection to different knapsack problems, e.g., there are approximation preserving reductions between the makespan scheduling problem if $q = 2$ and different variants of the multidimensional knapsack problem, which yields an FPTAS for $1|nr = 1, q = 2|C_{\max}$ (Györgyi and Kis [2]).

While the approximability of the makespan objective is quite well understood, see Györgyi and Kis [3], much less is known about the complexity and approximability of single and parallel machine scheduling under non-renewable resource constraints and the total weighted completion time objective. The complexity of the problem $1|nr = 1|\sum w_j C_j$ has been settled by Carlier [1], but no approximability results have been published since then, except for the special case with $q = 2$ supply dates for which Kis [5] provided an FPTAS. In fact, a major open question of the area is if there exists a constant-factor approximation algorithm for $1|nr = 1|\sum w_j C_j$ or not.

Table 1: New complexity and approximability results for special cases of $1|nr = 1|\sum w_j C_j$.

Row	**#Supp.** q	**Restriction**	**Objective function**	**Result**
1	$*$	$p_j = \bar{p}$, $a_j = \bar{a}$	$\sum w_j C_j$	polynomial time (decr. w_j ord.)
2	$*$	$p_j = \bar{p}$, $w_j = \bar{w}$	$\sum \bar{w} C_j$	polynomial time (incr. a_j ord.)
3	$*$	$a_j = \bar{a}, w_j = \lambda p_j$	$\sum w_j C_j$	polynomial time (decr. p_j ord.)
4	2	$p_j = 1$, $w_j = \lambda a_j$	$\sum w_j C_j$	weakly NP-hard
5	2	$w_j = p_j = a_j$	$\sum p_j C_j$	weakly NP-hard
6	$*$	$w_j = p_j = a_j$	$\sum p_j C_j$	strongly NP-hard
7	$*$	$w_j = p_j = a_j$	$\sum p_j C_j$	2-approx algorithm (LPT rule)
8	*const.*	$w_j = p_j$	$\sum p_j C_j$	PTAS
9	*const.*	$a_j = \bar{a}, w_j = 1$	$\sum C_j$	FPTAS
10	2	$p_j = 1$, $a_j = w_j$	$\sum w_j C_j$	2-approximation (decr. w_j ord.)
11	$*$	$p_j = 1$, $a_j = w_j$	$\sum w_j C_j$	3-approximation (decr. w_j ord.)

2 New results

In order to get more tractable variants of $1|nr = 1|\sum w_j C_j$, we introduce strong relations between various job parameters, like $a_j = w_j$, or $p_j = 1$, etc. In Table 1 we summarize our results. Each row represents a different variant: A 2 or *const* in the column q means that there are 2 or a constant number of supply dates, while if there is a star $*$, then q is arbitrary.

The first three rows correspond to polynomially solvable special cases. In all three cases the optimal solution can be obtained by scheduling the jobs in the specified order, while making delays only if there is not enough non-renewable resource left to start the next job. The proofs are based on simple interchange arguments in the first two cases, while in the third case some case distinction is needed.

The rows 4-6 provide new hardness results, which are somewhat surprising. For instance, if each job j has only a single parameter, the processing time p_j, and $a_j = w_j = p_j$, then $1|nr = 1, p_j = a_j = w_j| \sum w_j C_j$ is strongly NP-hard, and it remains NP-hard even if the number of supply dates is $q = 2$.

In rows 7-8 we summarize two approximability results when $w_j = p_j$ for each job j. If, in addition, $a_j = p_j$ also holds, then for arbitrary number of supply dates, scheduling the jobs in non-increasing processing time order is a 2-approximation algorithm. However, the proof of this fact is rather involved. Assuming only $w_j = p_j$ for each job j, while q is a constant, we were able to devise a PTAS, but the proof needs some new ideas. The above results are from Györgyi and Kis [4].

In row 9 we have the case where there is no connection between the processing times and the weights of the jobs, but each demand is the same and there is a constant number of supply dates. For this variant there exists an FPTAS.

Finally, in rows 10-11, we have $p_j = 1$, and $a_j = w_j$. In fact, this variant is NP-hard, see row 4. We schedule the jobs in non-increasing p_j order. If $q = 2$, then we could show that it yields a 2-approximation algorithm, but if $q \geq 3$, then we could show only that the same algorithm yields a 3-approximation. However, the proofs, especially of the former case, are not straightforward.

References

[1] J. Carlier (1984). *Problèmes d'ordonnancements à contraintes de ressources: algorithmes et complexité*. Thèse d'état.

[2] P. Györgyi, T. Kis (2015). *Reductions between scheduling problems with non-renewable resources and knapsack problems*, Theoretical Computer Science, 565, 63–76.

[3] P. Györgyi, T. Kis (2017). *Approximation schemes for parallel machine scheduling with non-renewable resources*, European J. of Operational Research, 258, 113–123.

[4] P. Györgyi, T. Kis (2019). *Minimizing total weighted completion time on a single machine subject to non-renewable resource constraints*, Journal of Scheduling, accepted.

[5] T. Kis (2015). *Approximability of total weighted completion time with resource consuming jobs*, Operations Research Letters, 43(6), 595–598.