Minimizing the total weighted completion time in single machine scheduling with non-renewable resource constraints

Péter Györgyi[1], Tamás Kis[1]

Institute for Computer Science and Control, Budapest, Hungary
gyorgyi.peter@sztaki.mta.hu, kis.tamas@sztaki.mta.hu

Keywords: single machine scheduling, non-renewable resources, total weighted completion time.

1 Introduction

In a machine scheduling problem with non-renewable resources, besides the machine(s), there are non-renewable resources, like raw materials, energy, or money, consumed by the jobs. The non-renewable resources have some initial stock, and they are replenished over time in given quantities. The objective function can be any of the widely-used optimization criteria in machine scheduling problems, see e.g., Carlier (1984) or Györgyi and Kis (2017).

Now, we consider a single machine variant with a single non-renewable resource. Formally, there is 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 resource requirement $a_j \geq 0$. The non-renewable resource has an initial stock $\tilde{b}_1 \geq 0$ at time $u_1 = 0$, and it is replenished at $q-1$ distinct supply dates $0 < u_2 < \cdots < u_q$ in quantities $\tilde{b}_\ell \geq 0$ for $\ell = 2, \ldots, q$. However, the total demand does not exceed the total supply, i.e., $\sum_{j \in \mathcal{J}} a_j \leq \sum_{\ell=1}^{q} \tilde{b}_\ell$. The cumulative supply up to supply date u_ℓ is $b_\ell = \sum_{k=1}^{\ell} \tilde{b}_k$. A schedule specifies the starting time S_j of each job $j \in \mathcal{J}$; it is feasible if (i) no pair jobs overlap in time, i.e., $S_{j_1} + p_{j_1} \leq S_{j_2}$ or $S_{j_2} + p_{j_2} \leq S_{j_1}$ for each pair of distinct jobs j_1 and j_2, 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 $\sum_{j \in J : S_j \leq t} a_j \leq b_\ell$.

An example problem along with a feasible schedule is depicted in Figure 1. There are 5 jobs represented by 5 rectangles. For each job j, the width of the corresponding rectangle indicates its processing time, while the resource requirement a_j is provided in the rectangle. Further on, there is an initial supply of $\tilde{b}_1 = 3$ at time $u_1 = 0$, and two more supplies at u_2 and u_3 with supplied quantities $\tilde{b}_2 = 4$ and $\tilde{b}_3 = 6$, respectively. In the depicted schedule, job j_1 cannot start earlier, since it requires 2 units from the resource, but there is only $\tilde{b}_1 + \tilde{b}_2 - a_2 - a_5 - a_3 = 1$ unit on stock before the supply arrives at u_3.

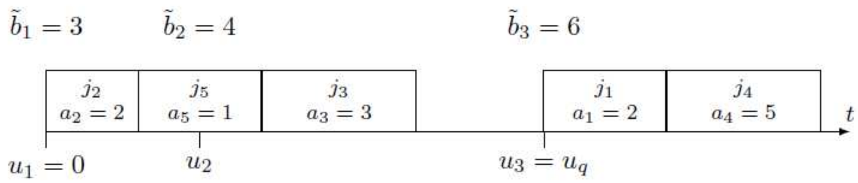

Fig. 1. A feasible schedule ($n = 5$, $q = 3$)

We aim at finding a feasible schedule S minimizing the total weighted completion time $\sum_{j\in\mathcal{J}} w_j C_j$, where $C_j := S_j + p_j$ denotes the completion time of job j. Using the standard $\alpha|\beta|\gamma$ notation, we denote our problem by $1|nr=1|\sum w_j C_j$, where '$nr=1$' indicates that we have only one type of non-renewable resource.

1.1 Previous results

The first results of the area are from the 1980s. Carlier (1984) presented several complexity results for variants where the makespan, the maximum lateness, or the total completion time have to be minimized in single and parallel machine environments. Slowinski (1984) examined a preemptive version of the problem for parallel machines. Toker *et. al.* (1991) and Xie (1997) applied reductions to the two-machine flow shop problem for variants where the supplies arrive uniformly over time. Grigoriev *et. al.* (2005) presented easy approximation algorithms for the makespan and the lateness objective. Gafarov *et. al.* (2011) proved several complexity results for various objective functions. Györgyi and Kis (2014) presented approximation schemes for the makespan objective in case of one resource. This was extended for a constant number of resources by Györgyi and Kis (2015b) and for parallel machines by Györgyi and Kis (2017) and by Györgyi (2017). Györgyi and Kis (2015a) proved reductions between the makespan minimization problem with two supply dates and variants of the Knapsack Problem. The most relevant antecedent of this research is Kis (2015), which considered the same objective function and presented an FPTAS for the problem with $q=2$.

1.2 Preliminaries

This paper examines variants with more supplies, where we can state job independent connections among the processing times, the resource requirements and the weights. If these connections are strong enough we can find easy ordering rules that yield optimal schedules, see Table 1. In the next sections we deal with two other variants.

Table 1. Easy variants of $1|nr=1|\sum w_j C_j$.

Variant	Optimal schedule
$p_j = a_j = \bar{a}$	non-increasing w_j order
$p_j = w_j = 1$	non-decreasing a_j order
$a_j = w_j = 1$	SPT order
$w_j = \bar{w},\ p_j = a_j$	SPT order
$a_j = \bar{a},\ p_j = w_j$	LPT order

Notice that SPT and LPT means that jobs are ordered in increasing, respectively, decreasing processing time order. In the corresponding algorithm, jobs are simply scheduled in increasing (SPT) / decreasing (LPT) processing time order. If the resource level is below the requirement of the next job, we simply wait until enough supply arrives.

While the SPT order gives the optimal schedule for the problem $1||\sum C_j$ (all job weights are 1), the LPT order is originally used in a list scheduling algorithm for the parallel machine problem $P||C_{\max}$ where it yields a 4/3-approximation algorithm.

2 The problem $1|nr=1, p_j = a_j = w_j|\sum w_j C_j$

Surprisingly, this very restrictive case is already NP-hard:

Theorem 1. *The problem* $1|nr = 1, q = 2, p_j = a_j = w_j| \sum w_j C_j$ *is weakly NP-hard, and* $1|nr = 1, p_j = a_j = w_j| \sum w_j C_j$ *is strongly NP-hard.*

These complexity results are new, formerly only the NP-hardness of the variant $1|nr = 1, q = 2| \sum C_j$ (see Kis (2015)) and that of $1|nr = 1| \sum C_j$ (Carlier (1984), Kis (2015)) were known.

However, we could derive a 2-approximation algorithm for it.

Theorem 2. *Scheduling the jobs in LPT order is a 2-approximation algorithm for* $1|nr = 1, p_j = a_j = w_j| \sum w_j C_j$.

3 A PTAS for $1|nr = 1, p_j = w_j, q = const| \sum w_j C_j$

In this section we describe an PTAS (polynomial time approximation scheme) for $1|nr = 1, p_j = w_j, q = const| \sum w_j C_j$. Notice that the resource consumption of the jobs is job-dependent, but the number of supplies is a constant, not part of the input. A PTAS is a family of algorithms $\{A_\varepsilon\}_{\varepsilon>0}$, such that for each $\varepsilon > 0$, A_ε is an $(1 + \varepsilon)$-approximation algorithm for the problem with a complexity polynomially bounded in the size of the input.

Let $P_{\text{sum}} := \sum_j p_j$ be the total processing time of the jobs. Let $\Delta := 1 + (\varepsilon/q^2)$. We will guess the total processing time of those jobs scheduled after u_ℓ for $\ell = 2, \ldots, q$, where a guess is a $q - 1$ dimensional vector of non-increasing numbers $P_2^g, \ldots, P_q^g$, i.e., $P_\ell^g \geq P_{\ell+1}^g \geq 1$ for $\ell = 2, \ldots, q - 1$, and each P_ℓ^g is of the form Δ^t for some integer $t \geq 0$ with $\Delta^t \leq P_{\text{sum}}$. Also fix $P_1^g := P_{\text{sum}}$. For any guess, define the set of *medium size jobs* $\mathcal{M}_\ell := \{j \mid p_j \geq (\Delta - 1)P_\ell^g\}$. Note that $\mathcal{M}_q \supseteq \mathcal{M}_{q-1} \supseteq \cdots \supseteq \mathcal{M}_1$, since $P_q^g \leq P_{q-1}^g \leq \cdots \leq P_1^g$. Let $\mathcal{S}_\ell$ be the complement of $\mathcal{M}_\ell$, i.e., $\mathcal{S}_\ell := \{j \mid p_j < (\Delta - 1)P_\ell^g\}$. Clearly, $\mathcal{S}_q \subseteq \mathcal{S}_{q-1} \subseteq \cdots \subseteq \mathcal{S}_1$. After these preliminaries, the PTAS for $1|nr = 1, p_j = w_j, q = const| \sum w_j C_j$ consists of the following steps:

1. Consider each possible guess $(P_2^g, \ldots, P_q^g)$ of the total processing time of those jobs starting after the supply dates $u_2, \ldots, u_q$, respectively. For each possible guess define the sets of jobs $\mathcal{M}_\ell$ and $\mathcal{S}_\ell$ (see above), and perform the steps 2-5. After processing all the guesses, go to Step 6.
2. For each $\ell = 1, \ldots, q$, choose at most $1/(\Delta - 1)$ medium size jobs from $\mathcal{M}_\ell$ (since the sets $\mathcal{M}_\ell$ are not disjoint, care must be taken to choose each job at most once). For each possible choice $(T_1, \ldots, T_q)$ of the medium size jobs (where $T_\ell \subseteq \mathcal{M}_\ell$), perform steps 3-5. After evaluating all choices, continue with the next guess in Step 1.
3. Determine a schedule of the medium jobs. That is, for $\ell = q, \ldots, 2$, schedule the jobs in T_ℓ in any order after u_ℓ contiguously, and if necessary, push to the right the jobs in $\bigcup_{\ell'=\ell+1}^q T_{\ell'}$.
4. Let $\mathcal{J}_0^u$ be the set of unscheduled jobs. For $\ell = q, q - 1, \ldots, 1$, repeat the following. In a general step with $\ell \geq 2$, pick jobs from $\mathcal{J}_{q-\ell}^u \cap \mathcal{S}_\ell$ in non-increasing a_j/p_j order until the selected subset K_ℓ satisfies $p(K_\ell) + p(T_\ell) \geq P_\ell^g - (1/\Delta)P_{\ell+1}^g$, or if no more jobs left, $K_\ell = \mathcal{J}_{q-\ell}^u \cap \mathcal{S}_\ell$. In either case, insert the jobs of K_ℓ in any order after u_ℓ and after all the jobs in $T_1 \cup \cdots \cup T_{\ell-1}$, and before all the jobs in $T_\ell \cup \bigcup_{\ell'=\ell+1}^q (K_{\ell'} \cup T_{\ell'})$ (pushing some of them to the right if necessary). Let $\mathcal{J}_{q-\ell+1}^u := \mathcal{J}_{q-\ell}^u \setminus K_\ell$ and continue with $\ell - 1$ until $\ell = 1$ or no more unscheduled jobs are left. For $\ell = 1$ just schedule all the remaining jobs from time $u_1 = 0$ on (pushing the already scheduled jobs to the right, if necessary). If the complete schedule obtained satisfies the resource constraints, then continue with Step 5, otherwise with the next choice of medium size jobs in Step 2.
5. Compute the objective function value of the complete schedule obtained in step (4), and store this schedule as the best schedule if it is the first feasible schedule or if it is better than the best feasible schedule found so far. Continue with next choice of medium size jobs in Step 2.

6. Output the best schedule found in the previous steps.

Theorem 3. *The proposed algorithm is an PTAS for* $1|nr=1, p_j = w_j, q = const|\sum w_j C_j$.

Acknowledgements

This work has been supported by the National Research, Development and Innovation Office - NKFIH grant K112881, and by the GINOP-2.3.2-15-2016-00002 grant of the Ministry of National Economy of Hungary.

References

Carlier J., 1984, "Problèmes d'ordonnancement à contraintes de ressources: algorithmes et complexité", *Université Paris VI-Pierre et Marie Curie, Institut de programmation*

Gafarov E.R., A.A. Lazarev and F. Werner, 2011, "Single machine scheduling problems with financial resource constraints: Some complexity results and properties", *Math. Social Sci.*, Vol. 62, pp. 7-13.

Grigoriev A., M. Holthuijsen and J. van de Klundert, 2005, "Basic scheduling problems with raw material constraints", *Naval Res. Logist.*, Vol. 52, pp. 527-553.

Györgyi P., 2017, "A PTAS for a resource scheduling problem with arbitrary number of parallel machines", *Oper. Res. Lett.*, Vol. 45, pp. 604-609.

Györgyi P., T. Kis, 2014, "Approximation schemes for single machine scheduling with non-renewable resource constraints", *J. Sched.*, Vol. 17, pp. 135-144.

Györgyi P., T. Kis, 2015a, "Reductions between scheduling problems with non-renewable resources and knapsack problems", *Theoret. Comput. Sci.*, Vol. 565, pp. 63-76.

Györgyi P., T. Kis, 2015b, "Approximability of scheduling problems with resource consuming jobs", *Ann. Oper. Res.*, Vol. 235, pp. 319-336.

Györgyi P., T. Kis, 2017, "Approximation schemes for parallel machine scheduling with non-renewable resources", *European J. Oper. Res.*, Vol. 258, pp. 113-123.

Kis T., 2015, "Approximability of total weighted completion time with resource consuming jobs", *Oper. Res. Lett.*, Vol. 43, pp. 595-598.

Slowinski R., 1984, "Preemptive scheduling of independent jobs on parallel machines subject to financial constraints", *European J. Oper. Res.*, Vol. 15, pp. 366-373.

Toker A., S. Kondakci and N. Erkip, 1991, "Scheduling under a non-renewable resource constraint", *J. Oper. Res. Soc.*, Vol. 42, pp. 811-814.

Xie J., 1997, "Polynomial algorithms for single machine scheduling problems with financial constraints", *Oper. Res. Lett.*, Vol. 21, pp. 39-42.