Joint replenishment meets scheduling

Tamás Kis (Speaker) [*] Péter Györgyi [†] Tímea Tamási [‡]
József Békési [§]

1 Introduction

We consider a combination of two classic problems of operations research: the joint replenishment problem (JRP) and single machine scheduling with release dates. In this problem, each job requires some resources and it can be processed on the machine after the required resources are replenished. One has to decide both about the replenishments and the schedule of the machine. The first subproblem is a well-known variant of the joint replenishment problem, while the second one belongs to the class of single machine scheduling problems with release dates, where the release dates are determined by the replenishment times of the required resources.

In the joint replenishment problem, the goal is to fulfill a set of demands (jobs) emerging over the time horizon. A demand can be fulfilled by ordering its required items (resources) not sooner than the arrival time of the demand. Orders of different demands can be combined, and the cost of simultaneously ordering a subset of item types incurs a joint ordering cost and an additional item ordering cost for each item type in the order. None of these costs depends on the number of units ordered. One of the main variants of this problem is the so-called *JRP-W*, where the objective is to minimize the sum of the ordering costs and the cost incurred by delaying the fulfillment of the jobs. Our problem is an extension of this variant: each demand (job) has to be processed on a single machine after the required items (resources) are replenished. The objective is to minimize the total ordering cost plus a scheduling criterion. We provide several complexity results for the offline problem, and competitive analysis for online variants with min-sum and min-max criteria, respectively.

Formally, we have a set $\mathcal{J}$ of n jobs that have to be scheduled on a single machine. Each job j has a processing time $p_j > 0$, a release date $r_j \geq 0$, and

[*] kis.tamas@sztaki.hu. Institute for Computer Science and Control, Eötvös Loránd Research Network, Kende Str. 13-17., 1111 Budapest, Hungary.

[†] gyorgyi.peter@sztaki.hu. Institute for Computer Science and Control, Eötvös Loránd Research Network.

[‡] tamasi.timea@sztaki.hu. Institute for Computer Science and Control, Eötvös Loránd Research Network and Department of Operations Research, Eötvös Loránd University.

[§] bekesi@inf.u-szeged.hu. Department of Computer Algorithms and Artificial Intelligence, Faculty of Science and Informatics, University of Szeged, Hungary.

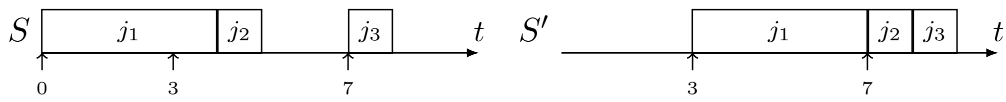

Figure 1: Two feasible solutions. The arrows below the time axis indicate the replenishment time points.

possibly a weight $w_j > 0$ (in case of min-sum type objective functions). In addition, there is a set of resources $\mathcal{R} = \{R_1, ..., R_s\}$, and each job $j \in \mathcal{J}$ requires a non-empty subset $R(j)$ of $\mathcal{R}$. A job j can only be started if all the resources in $R(j)$ are replenished after r_j. Each time some resource R_i is replenished, a fixed cost K_i is incurred on top of a fixed cost K_0, which must be paid each time any replenishment occurs. These costs are independent of the replenished amount.

A solution of the problem is a pair $(S, \mathcal{Q})$, where S is a schedule specifying a starting time for each job $j \in \mathcal{J}$, and $\mathcal{Q} = \{(\mathcal{R}_1, t_1), \ldots, (\mathcal{R}_q, t_q)\}$ is a replenishment structure, which specifies time moments t_ℓ along with subsets of resources $\mathcal{R}_\ell \subseteq \mathcal{R}$ such that $t_1 < \ldots < t_q$. We say that job j is *ready to be started at time moment t with respect to replenishment structure* $\mathcal{Q}$, if each resource $R \in R(j)$ is replenished at some time moment in $[r_j, t]$, i.e., $R(j) \subseteq \cup_{t_\ell \in [r_j, t]} \mathcal{R}_\ell$. The solution is *feasible* if (i) the jobs do not overlap in time, i.e., $S_j + p_j \leq S_k$ or $S_k + p_k \leq S_j$ for each $j \neq k$, and (ii) each job $j \in \mathcal{J}$ is ready to be started at S_j w.r.t. $\mathcal{Q}$.

The *cost of a solution* is the sum of the scheduling cost c_S, and the replenishment cost $c_{\mathcal{Q}}$. The former can be any optimization criteria know in scheduling theory (e.g., the total weighted completion time $\sum w_j C_j$, the total flow time $\sum F_j$, or the maximum flow time $F_{\max}$). The replenishment cost is calculated as follows: $c_{\mathcal{Q}} := \sum_{\ell=1}^{|\mathcal{Q}|} (K_0 + \sum_{R_i \in \mathcal{R}_\ell} K_i)$.

Example 1 *Suppose there are 3 jobs, $p_1 = 4$, $p_2 = p_3 = 1$, $r_1 = 0$, $r_2 = 3$, and $r_3 = 7$, and the objective is to minimize $\sum C_j$. The replenishment costs are K_0 and K_1, and we deliberately do not specify numerical values for these parameters. If there are 3 replenishments from a single resource R, i.e., $\mathcal{Q} = ((\{R\}, 0), (\{R\}, 3), (\{R\}, 7))$, and the starting times of the jobs are $S_1 = 0, S_2 = 4$, and $S_3 = 7$, then $(S, \mathcal{Q})$ is a feasible solution with total replenishment cost of $3(K_0 + K_1)$ and total completion time 17 (Figure 1 left).*

However, if there are only two replenishments in $\mathcal{Q}'$ at $t_1 = 3$ and $t_2 = 7$, then we have to start the jobs later, e.g., if $S'_1 = 3, S'_2 = 7$, and $S'_3 = 8$, then $(S', \mathcal{Q}')$ is feasible. Observe that in the second solution we have saved the cost of a replenishment $(K_0 + K_1)$, however, the total completion time of the jobs has increased from 17 to 24 (see Figure 1 right).

As to whether $(S, \mathcal{Q})$ is better than $(S', \mathcal{Q}')$ or not depends on the value of $K_0 + K_1$.

Table 1: Results of the paper.

Problem	Result
$1\|jrp, s=1, r_j\|\sum C_j + c_Q$	NP-hard
$1\|jrp, p_j=1, r_j\|\sum C_j + c_Q$	NP-hard
$1\|jrp, s=2, r_j\|F_{\max} + c_Q$	NP-hard
$1\|jrp, s=const, p_j=1, r_j\|\sum w_j C_j + c_Q$	polynomial alg.
$1\|jrp, s=const, p_j=p, r_j\|\sum C_j + c_Q$	polynomial alg.
$1\|jrp, s=1, r_j\|F_{\max} + c_Q$	polynomial alg.
$1\|jrp, s=const, p_j=p, r_j\|F_{\max} + c_Q$	polynomial alg.
$1\|jrp, s=1, p_j=1, r_j\|\sum C_j + c_Q$	2-competitive alg.
$1\|jrp, s=1, p_j=1, r_j\|\sum C_j + c_Q$	no $\left(\frac{3}{2}-\varepsilon\right)$-competitive alg.
$1\|jrp, s=1, p_j=1, r_j\|\sum w_j C_j + c_Q$	no $\left(\frac{\sqrt{5}+1}{2}-\varepsilon\right)$-competitive alg.
$1\|jrp, s=1, p_j=1, r_j\|\sum F_j + c_Q$	2-competitive alg.
$1\|jrp, s=1, p_j=1, r_j\|\sum F_j + c_Q$	no $\left(\frac{3}{2}-\varepsilon\right)$-competitive alg.
$1\|jrp, s=1, p_j=1, regular\ r_j\|F_{\max} + c_Q$	$\sqrt{2}$-competitive alg.
$1\|jrp, s=1, p_j=1, regular\ r_j\|F_{\max} + c_Q$	no $\left(\frac{4}{3}-\varepsilon\right)$-competitive alg.
$1\|jrp, s=1, p_j=1, r_j\|F_{\max} + c_Q$	no $\left(\frac{\sqrt{5}+1}{2}-\varepsilon\right)$-competitive alg.

* jrp indicates the joint replenishment extension, $s=1$ limits the number of resources to 1, and c_Q is the total replenishment cost.

2 Results

The main results of the paper fall in 3 categories: (i) NP-hardness proofs, (ii) polynomial time algorithms, and (iii) competitive analysis of online variants of the problem, see Table 1 for an overview. We provide an almost complete complexity classification for the offline problems with both of the $\sum w_j C_j$ and $F_{\max}$ objectives. Notice that the former results imply analogous ones for the $\sum w_j F_j$ criterion. While most of our polynomial time algorithms work only with unit-time jobs, a notable exception is the case with a single resource and the $F_{\max}$ objective, where the job processing times are arbitrary positive integer numbers. We have devised online algorithms for some special cases of the problem for both min-sum and min-max criteria. In all variants for which we present an online algorithm with constant competitive ratio, we have to assume unit-time jobs. While we have a 2-competitive algorithm with unit time jobs for min-sum criteria, for the online problem with the $F_{\max}$ objective we also have to assume that the input is regular, i.e., in every time unit a new job arrives, but in this case the competitive ratio is $\sqrt{2}$. The technical details can be found in [1].

Acknowledgment

This work has been supported by the National Research, Development and Innovation Office grants no. TKP2021-NKTA-01, and ED 18-2-2018-0006. The research of Péter Györgyi was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

[1] P. Györgyi, T. Kis, T. Tamási, J. Békési (2021). *Joint replenishment meets scheduling.* https://arxiv.org/abs/2104.09178