Algorithms and complexity results for resource leveling problems

Bendotti Pascale[12], Brunod Indrigo Luca[2], Chrétienne Philippe[2], Escoffier Bruno[23]

[1] EDF R&D, 7 boulevard Gaspard Monge, 91120 Palaiseau, France

[2] Sorbonne Université, CNRS, LIP6 UMR 7606, 4 place Jussieu, 75005 Paris, France

[3] Institut Universitaire de France

Keywords : *scheduling, resource leveling, complexity*

1 Introduction

In many scheduling problems, jobs require resources such as workers or specific equipment in order to be executed. One can suppose that the amount of a resource available at any given time is limited and therefore imposes a hard constraint such as in the widely studied RCPSP. In practice however, it is often possible to obtain additional resources when needed but at a significant cost, by hiring extra workforce for instance. The field of scheduling known as resource leveling aims to obtain regularity in resource use by modeling such costs. The literature on this subject (see [4] for a recent survey) uses different leveling objective functions and mostly provides heuristics and MIP-based approaches.

The aim of this work is to study resolution methods and complexity for resource leveling problems under classical scheduling constraints.

2 Problem

Consider a project with a set of jobs J and a single renewable resource with resource level $L \in \mathbb{N}$. Each job $i \in J$ has processing time $p_i \in \mathbb{R}_+$ and resource consumption $c_i \in \mathbb{N}$.

In a given schedule $x \in \mathbb{R}_+^J$, the resource consumption at time $t \in \mathbb{R}_+$ is the sum of c_i for i being processed at t, namely $t \mapsto \sum_{i \in J} \mathbb{1}_{[x_i, x_i+p_i)}(t) c_i$. Classically, a resource level L is associated with a constraint that resource consumption never exceeds L. Here, L can be exceeded, if it is necessary to meet a constraint, but the objective function penalizes such resource overspending.

A natural choice for the objective is to minimize the resource consumption exceeding the level or to maximize resource consumption that fits under the level, which is equivalent for exact resolution. The function formally writes:

$$F(x) = \int_{t=0}^{+\infty} \min\left(L, \sum_{i \in J} \mathbb{1}_{[x_i, x_i+p_i)}(t) c_i\right) dt$$

Function F is illustrated in Figure (1) on an instance with four jobs. The area corresponding to its value is shown in gray.

In order to investigate various resource leveling problems, a few classical scheduling constraints were selected and, for each of them, the complexity was studied for fixed values of parameters L, c_i and p_i and for more general cases.
The considered constraints are the following:

- Makespan bound: constraining the makespan of the project to be lower than a given bound M is a first way to obtain non-trivial resource leveling problems, not to mention that such a constraint makes a lot of sense in practice;

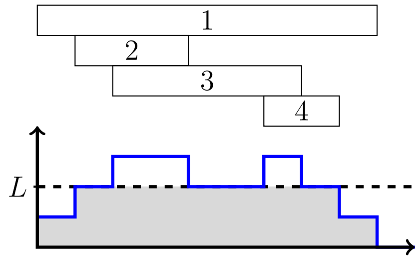

FIG. 1: Illustration of function F.

- Release and due dates: each job must be executed within a certain time interval, both preemptive and non-preemptive cases were studied;
- Precedence constraints: some jobs must be completed before others start, the particular case of a tree precedence graph was considered, as well as the general case.

As for parameters, the values $L = 1$, $L = 2$ and $L \in \mathbb{N}$ were considered, each case being declined in sub-cases with unit resource consumptions ($c_i = 1$) and unit processing times ($p_i = 1$).

3 Results

Polynomial-time algorithms were found for several problems and, interestingly, some of them are inspired from classical scheduling algorithms. For example, an adaptation of Hu's algorithm [5] was found to be optimal for a tree precedence graph and unit processing times. Similarly, the case of $L = 2$ and unit processing times was solved using the result of [2] for two-machine scheduling and relying on the same kind of matching structures.

Some NP-hardness and approximation results were found for other problems, again with strong inspiration from classical scheduling literature. For the case $L = 1$, reductions from 3-Partition use the same constructions as single-machine scheduling complexity proofs [3]. Moreover, the NP-hardness of resource leveling problems with unit resource consumption was proven using a reduction directly from machine scheduling problems of known complexity [1, 6].

All those similarities in both resolution methods and complexity show that strong links exist between resource leveling problems and some of their classical scheduling counterparts that share the same constraints. Those links however are sometimes far from obvious, leveling function F being fundamentally different from standard scheduling objectives.

References

[1] J. Du, J.Y.-T. Leung, and G.H. Young. Scheduling chain-structured tasks to minimize makespan and mean flow time. *Inform. and Comput.*, 92(2):219–236, 1991.

[2] M Fujii, T Kasami, and K Ninomiya. Optimal sequencing of two equivalent processors. *SIAM Journal on Applied Mathematics*, 17(4):784–789, 1969.

[3] Michael R Garey and David S Johnson. Computers and intractability. *A Guide to the*, 1979.

[4] Sönke Hartmann and Dirk Briskorn. An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of operational research*, 297(1):1–14, 2022.

[5] Te C Hu. Parallel sequencing and assembly line problems. *Operations research*, 9(6):841–848, 1961.

[6] J.D. Ullman. NP-complete scheduling problems. *J. Comput. System Sci.*, 10:384–393, 1975.