Learning based heuristics for scheduling jobs with release dates on a single machine to minimize the sum of completion times

Axel Parmentier[1], Vincent T'kindt[2]

[1] CERMICS, Ecole des Ponts, Marne-la-Vallée, France
axel.parmentier@enpc.fr
[2] University of Tours,
LIFAT (EA 6300), ERL CNRS ROOT 7002, Tours, France
tkindt@univ-tours.fr

Mots-clés : *scheduling, structured machine learning.*

1 Introduction

Consider the problem where n jobs have to be scheduled on a single machine. Each job j is defined by a *processing time* p_j and a release date r_j so that, in a given schedule, no job j can start before its release date. The machine can only process one job at a time and preemption is not allowed. The goal is to find a schedule s (permutation) that minimizes the total completion time $\sum_j C_j(s)$ with $C_j(s)$ the completion time of job j in schedule s. If $s = (j_1, \ldots, j_n)$, then

$$C_{j_1}(s) = r_{j_1} + p_{j_1} \quad \text{and} \quad C_{j_k}(s) = \max(C_{j_{k-1}}(s), r_{j_k}) + p_{j_k} \text{ for } k > 1.$$

When there is no ambiguity, we omit the reference to schedule s when referring to completion times. Following the standard three-field notation in scheduling theory, this problem is referred to as $1|r_j|\sum_j C_j$ and is strongly $\mathcal{NP}$-hard [5]. When there is no release dates, the corresponding $1||\sum_j C_j$ problem can be solved in $O(n \log(n))$ time using the SPT rule (shortest processing times first).

The $1|r_j|\sum_j C_j$ problem is a challenging problem which has been studied for a long time. In this work, we focus on heuristic algorithms which can be used to compute good solutions in a reasonable amount of time. Along the years, numerous heuristic algorithms have been proposed. We cite the RDI (Release Date Improvement procedure) local search based on the APRTF (Advanced Priority Rule for Total Flowtime) greedy rule proposed in [2] and which requires $O(n^4 \log(n))$ time. The RBS heuristic (Recovering Beam Search) developed in [4] is a truncated search tree approach that has been the heuristic with the best performances for a decade. The RBS heuristic requires $O(wn^3 \log(n))$ time with w the beam width parametrizing the heuristic : notice that in [4] the case $w = 1$ is considered. To the best of our knowledge, the state-of-the-art heuristic is a matheuristic proposed in [3] which provides solutions very close to the optimal ones but at the price of a large CPU time requirement.

In this work we propose learning based heuristics for the $1|r_j|\sum_j C_j$ problem which are compared to these milestones heuristics.

2 Learning based heuristics for scheduling

The use of machine learning (ML) techniques within operations research (OR) algorithms is a recent but active and promising research area [1]. To the best of our knowledge, very few contributions of this kind have considered scheduling problems. In these works, ML is used to guide the solution process, *i.e.*, the proposed OR heuristic. In this paper, we elaborate on an

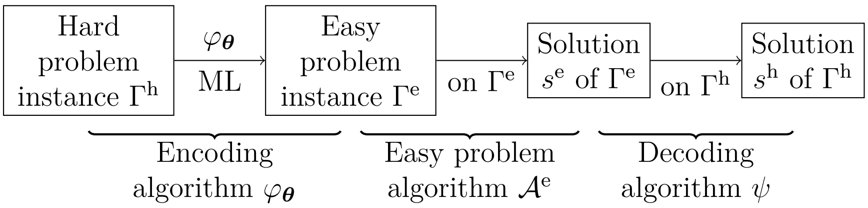

FIG. 1 – ML to approximate hard problems by well-solved ones

original approach recently introduced in [6] and illustrated in Figure 1. A ML predictor $\varphi_{\boldsymbol{\theta}}$, which we call the encoding algorithm, is used to convert an instance Γ^{h} of the *hard* $1|r_j|\sum_j C_j$ problem into an instance Γ^{e} of the $1||\sum_j C_j$ problem. The latter is called the *easy* problem as it exists a practically efficient algorithm $\mathcal{A}^{\mathrm{e}}$ to solve it (SPT rule). From an optimal solution s^{e} of Γ^{e}, a decoding algorithm is used to rebuild a solution s^{h} to Γ^{h}. Notice that, the processing time $\hat{p}_j$ of job j in Γ^{e} is not equal to its processing time p_j in Γ^{h}, but to a linear combination of features computed from Γ^{h}. Several different decoding algorithms are proposed to obtain a schedule for the $1|r_j|\sum_j C_j$ problem, thus leading to different heuristics.

The main challenge to make such an approach working is to build an encoding algorithm $\varphi_{\boldsymbol{\theta}}$ such that the optimal solution of the instance Γ^{e} leads to a good solution Γ^{h} after decoding. As usual in ML, we first define an appropriate family of predictors $(\varphi_{\boldsymbol{\theta}})_{\boldsymbol{\theta}}$, and then seek (learn) the best parameter $\boldsymbol{\theta}$. We formulate the choice of $\boldsymbol{\theta}$ as a structured learning problem. Structured learning algorithms on the permutation group have been thoroughly studied in the literature on rankings, but with applications such as document retrieval or question answering that are quite far from scheduling. If the traditional learning approaches of this literature can in theory be applied in our context, we do not use them because the loss functions they use to evaluate if a ranking is a good approximation of another are not good evaluations of the quality of a $1|r_j|\sum_j C_j$ schedule. We instead propose a novel generic approach based on Fenchel-Young loss functions which can be applied to a large set of hard problem [7].

Computational experiments show that the proposed learning based heuristics compete with the state-of-the-art heuristics and enable to efficiently solve very large instances.

Références

[1] Y. Bengio, A. Lodi and A. Prouvost. Machine Learning for Combinatorial Optimization : A Methodological Tour d'horizon. European Journal of Operational Research, 290(2) :405-421, 2021.

[2] S. Chand, R. Traub and R. Uzsoy. An iterative heuristic for the single machine dynamic total completion time scheduling problem. Annals of Operations Research, 23(7) :641-651, 1996.

[3] F. Della Croce, F. Salassa and V. T'kindt A hybrid heuristic approach for single machine scheduling with release times. Computers & Operations Research, 45 :7-11, 2014.

[4] F. Della Croce and V. T'kindt A recovering beam search algorithm for the one-machine dynamic total completion time scheduling problem. Journal of the Operational Research Society, 53 :1275-1280, 2002.

[5] A.H.C. Rinnooy Kan. Machine sequencing problem : classification, complexity and computation. Springer-Verlag, 1976.

[6] A. Parmentier. Learning to Approximate Industrial Problems by Operations Research Classic Problems. Operations Research, 2021.

[7] A. Parmentier and V. T'kindt Learning to solve the single machine scheduling problem with release times and sum of completion times. arXiv :2101.01082, 2021.