A learning based matheuristic to solve the two machine flowshop scheduling problem with sum of completion times

Vincent t'Kindt, Romain Raveaux

A learning based matheuristic to solve the two machine flowshop scheduling problem with sum of completion times

Vincent T'kindt[1,2], Romain Raveaux[1]

[1] Université de Tours, Laboratoire d'Informatique Fondamentale et Appliquée de Tours (LIFAT - EA 6300), 64 Avenue Jean Portalis, 37000 Tours, France
tkindt@univ-tours.fr; romain.raveaux@univ-tours.fr

[2] ERL CNRS 7002 Recherche Opérationnelle, Ordonnancement et Transport, 64 Avenue Jean Portalis, 37000 Tours, France

Mots-clés : *Scheduling, Flowshop, Matheuristic, Deep Learning.*

1 Introduction

Consider the problem where n jobs have to be scheduled on two machines organized in a flowshop setting. Each job j is defined by a *processing time* $p_{j,i}$ on machine $i = 1, 2$ and has to be processed first on machine 1 and next on machine 2. Each machine can only process one job at a time and preemption is not allowed. We restrict the search for a solution to the set of permutation schedules. Therefore, 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. 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 $F2||\sum_j C_j$ and is strongly $\mathcal{NP}$-hard [3].

The $F2||\sum_j C_j$ problem is a challenging problem which has been studied for a long time from both exact and heuristic point of views. In this work, we don't consider exact approaches. Along the years, several heuristic algorithms have been proposed and, to the best of our knowledge, the most efficient one is a matheuristic proposed in [2]. Matheuristics are local search algorithms which explore the neighborhood of an incumbent solution by solving a mathematical programming formulation of the problem. In this work we propose the developpment of a learning based predictor to improve the matheuristic proposed in [2].

2 A learning based matheuristic

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. The learning based matheuristic we propose aims at improving upon the current matheuristic, referred to as `MATH`, in [2].

A solution s of the $F2||\sum_j C_j$ problem can be seen as a sequence of jobs and let $s[k]$ be the job scheduled at position k. `MATH` heuristic proceeds by selecting, at each iteration, a *window of positions* $[r; r+h]$ so that only positions $\{r, r+1, ..., r+h\}$ are rescheduled by means of a mixed integer formulation of the problem. Said differently, when r and h are fixed, the problem is to reschedule jobs in positions r to $r+h$ while keeping fixed the partial schedule from position 1 to $(r-1)$ and from position $(r+h+1)$ to n. At each iteration, Della Croce et al. [2] suggest to randomly select a value r between 1 and $(n-h)$ with $h = 12$. The latter choice resulting from an experimental evaluation of the average efficiency of their algorithm.

The strengh of `MATH` relies on the genuine exploitation of an efficient mixed integer formulation of the problem. However, it has several flaws induced by the selection of the r and h values at each iterations. First, two runs on the same instance of the flowshop problem rarely provides the same solution which is induced by the random choice of r. Besides, the convergence towards a good solution can be rather slow as few windows $[r; r+h]$ lead in practice to an improvement of the incumbent solution. At last, the choice of $h = 12$ is debattable as sometimes lower values lead to the same results but faster and as it could be worth trying sometimes larger tractable values.

We propose to build a predictor to guide the `MATH` heuristic at each iteration in the choice of r and h. The learning problem can be formulated as follows : we could learn to predict for a given schedule s and values r and h, if a reoptimization would lead to an improvement or not. This learning problem is a classification problem.

3 Solution of the learning problem

A deep learning solution was adopted to create our predictor. Deep Learning is about finding a good data representation with respect to an objective thanks to a composition of functions. Composition means that complicated functions are combinations of smaller, simpler functions. Deep learning relies on data. As a consequence, `MATH` was instrumented to generate a data set $\mathcal{D} = \{x_m, y_m\}_{m=1}^{M}$ from random instances of the scheduling problem. x_m is a triple composed of a job sequence, r and h. y_m is a Boolean value such that $y_m = 1$ if the window $[r; r+h]$ leads to an improvement and $y_m = 0$ otherwise. This raw data are complex and not easy to handle by machine learning techniques. So, we decided to build an embedding function ϕ. Each x_m is projected into a vector space of dimension d thanks to the function $\phi(x_m) \in \mathbb{R}^d$. Key information about the sequence and the window are extracted by the function ϕ which can be seen as a smart preprocessing to feed the predictor with a meaningful representation of a sequence and a window. The predictor predicts whether or not a reoptimization in the window may lead to an improvement. The predictor is a function $p(\phi(x_m), \theta^*)$ where parameters $\theta^* \in \Theta$ are found by solving the following learning problem : $\theta^* = arg \min_{\theta \in \Theta} \sum_{(x_m, y_m) \in \mathcal{D}} l(p(\phi(x_m), \theta), y_i)$, l being a loss function gauging the error between the predicted value and the true value (y_m). The learning problem is solved by a well-established gradient descent algorithm called Adam [4].

Preliminary results are encouraging. 77% of the windows leading to an improvement are correctly predicted (True Positive). 57% of the non-improving windows are well predicted (True Negative). The predictor acts as a filter avoiding the thorough reoptimization of unfruitful windows without discarding too much the promising windows. These results should be consolidated with larger experiments.

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] F. Della Croce, A. Grosso and F. Salassa. A matheuristic approach for the two-machine total completion time flow shop problem. Annals of Operation Research, 213 :67-78, 2004.

[3] M.R. Garey, D.S. Johnson and R. Sethi. The complexity of flowshop and jobshop scheduling. Mathematics of Operations Research, 1 :117-129, 1976.

[4] Kingma, D. & Ba, J. Adam : A Method for Stochastic Optimization. *ArXiv E-prints*. pp. earXiv :1412.6980 (2014,12)