A New FPT Algorithm for Scheduling Dependant Tasks on Parallel Machines

Istenc Tarhan[1,2], Jacques Carlier[2], Claire Hanen[1,3], Antoine Jouglet[2], Alix Munier Kordon[1]

[1] Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
{Istenc.Tarhan,Claire.Hanen,Alix.Munier}@lip6.fr
[2] UTC, Heudiasyc,France
{Istenc.Tarhan,Jacques.Carlier,Antoine.Jouglet}@hds.utc.fr
[3] UPL, Université Paris Nanterre, F-92000 Nanterre,France

Mots-clés : *Scheduling, fixed-parameter tractable, release times and deadlines, branch and bound, parallel machines*

1 Introduction

Scheduling problems with resource limitation and precedence constraints have many applications in various fields, such as production systems, the use of multi-core parallel machines or the design of embedded systems. Also, many authors have developed exact or approximate algorithms to efficiently solve these problems since the beginning of the sixties.

Our study considers a parallel machine scheduling problem defined by a set of non-preemptive jobs to be executed by identical machines. Each job has an integer release time r_i, a processing time p_i and a deadline d_i. It has to be scheduled by one machine, each of which can process at most one job at a time. Lastly, there are precedence relations between certain pairs of jobs.This problem is denoted by $P|prec, r_j, d_j|\star$ using the Graham notation [4]. This problem is known to be NP-hard, see as example Garey and Johnson [3].

Branch-and-Bound methods are commonly considered for solving exactly scheduling problems. In the nineties, several authors developed Branch-and-Bound methods to handle the resource-constrained scheduling problem (RCPS in short), see Brucker et al. [1] for a survey. The Demeulemeester and Herroelen algorithm [2] is one of the most efficient Branch-and-Bound method to solve efficiently this class of problems without release time and deadline. To our knowledge, there is no theoretical evaluation of the worst-case complexity of this algorithm.

Parameterized complexity gives a new insight into the theoretical and practical difficulty of solving problems instances, depending on the value of some parameters. A fixed-parameter tractable (FPT in short) algorithm solves any instance of size n of the problem with parameter k in a time $\mathcal{O}(f(k) \times \text{poly}(n))$, where f is allowed to be a computable superpolynomial function and $\text{poly}(n)$ a polynome of n. However few results have been produced for scheduling problems. Van Bevern et al. [6] defined an FPT algorithm for the RCPS problem parameterized by the tuple $(w(G), \lambda)$, where λ is the maximum allowed difference between the earliest starting time and factual starting time of a job and $w(G)$ is the width of the precedence graph. More recently, Munier and Hanen [5] developed a FPT algorithm for our problem with two parameters : a first parameter μ, called the pathwidth, which is the maximal number of overlapping jobs time windows (r_i, d_i) at a single time t and a second parameter, the maximal processing time $p_{\max}$ of a job.

This talk develops a new FPT algorithm inspired by Demeulemeester and Herroelen Branch-and-Bound method [2]. The idea here is to mix the efficiency in practice of this latter class of algorithms with the theoretical efficiency of FPT ones when the parameters are small. Our new Branch-and-Find (B&F in short) algorithm and the bound of complexity are briefly presented below.

2 The Branch-and-Find algorithm

The B&F algorithm is a search in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, instead of a tree, linking this algorithm to a dynamic programming approach. The definition of the nodes is similar than Demeulemeester and Herroelen algorithm [2] : a node is a quadruplet $v = (V, t, P, M) \in \mathcal{V}$ where V, P are job subsets with $P \subset V$, M defines the completion time of jobs of P and t is a time instant. The node v is then associated to a set of feasible s schedules such that :

1. Every job $i \in V \backslash P$ is completed before t : $s(i) + p_i \leq t$;
2. Every job in $i \in \mathcal{T} \backslash V$ starts after time t : $s(i) \geq t$;
3. Every job $i \in P$ such that $M_i = t$ starts at time $s(i) = M_i - p_i$. Every job $i \in P$ with $M_i > t$ starts either at time $M_i - p_i$ or at time $s(i) \geq t$.

Let $v = (V, t, P, M)$ and $v' = (V', t', P', M')$ be two nodes of $\mathcal{V}$. Roughly speaking, arc $(v, v') \in A$ if a feasible partial schedule associated to V' can be built from any partial schedule associated to v by adding one or several jobs performed at time t or a little bit later. Then, any longest path to a node $v_f = (V_f, t_f, P_f, M_f)$ with $V_f = \mathcal{V}$ defines a feasible semi-active schedule.

Our B&F algorithm simply develops graph $\mathcal{G}$ using several original dominance properties. The next theorem bounds its time complexity.

Theorem 1 *The B&F algorithm is an FPT algorithm parameterized by* (μ, p_{max}) *of time complexity in* $\mathcal{O}\left(n^3 \times \mu f(\mu, p_{max}) + n^2 \times h(\mu, p_{max}) + n \times g(\mu, p_{max}) \times h(\mu, p_{max})\right)$ *with the functions* $f(\mu, p_{max}) = \binom{2\mu}{\mu} \times p_{max}^{\mu} \times 2^{\mu}$, $g(\mu, p_{max}) = \mu \times (\mu \times \ln(\mu) + (2 \times p_{max})^{\mu})$ *and* $h(\mu, p_{max}) = \binom{\mu}{\lceil \mu/2 \rceil} \times f(\mu, p_{max})$.

Some experiments on random generated instances are also provided showing that the practical efficiency of our algorithm depends on the parameters but that the theoretical complexity is overestimated.

An exiting perspective of this work is to study the adaptation of other Branch-and-Bound based methods to FPT algorithm.

Références

[1] Peter Brucker, Andreas Drexl, Rolf Möhring, Klaus Neumann, and Erwin Pesch. Resource-constrained project scheduling : Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1) :3–41, 1999.

[2] Erik Demeulemeester and Willy Herroelen. A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management Science*, 38(12) :1803–1818, 1992.

[3] M.R. Garey and D.S. Johnson. Strong NP-completeness results : motivation, examples, and implications. *J. Assoc. Comput. Mach.*, 25(3) :499–508, 1978.

[4] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G.Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling : a survey. In P.L. Hammer, E.L. Johnson, and B.H. Korte, editors, *Discrete Optimization II*, volume 5 of *Annals of Discrete Mathematics*, pages 287–326. Elsevier, 1979.

[5] Claire Hanen and Alix Kordon Munier. Fixed-parameter tractability of scheduling dependent typed tasks subject to release times and deadlines. *Accepted to Journal of Scheduling*, 2022.

[6] René van Bevern, Robert Bredereck, Laurent Bulteau, Christian Komusiewicz, Nimrod Talmon, and Gerhard J. Woeginger. Precedence-constrained scheduling problems parameterized by partial order width. In Yury Kochetov, Michael Khachay, Vladimir Beresnev, Evgeni Nurminski, and Panos Pardalos, editors, *Discrete Optimization and Operations Research*, pages 105–120, Cham, 2016. Springer International Publishing.