A Fixed-Parameter Algorithm for Scheduling Unit dependent Tasks with Unit Communication Delays

Ning Tang[1], Alix Munier-Kordon[1]

Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

ning.tang@lip6.fr, Alix.Munier@lip6.fr

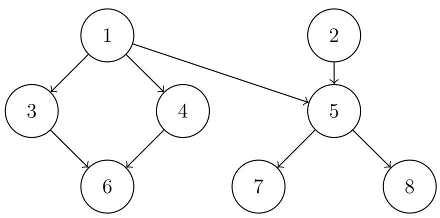
Mots-clés : *scheduling, communication delays, makespan, fixed-parameter algorithm.*

1 Problem description and related work

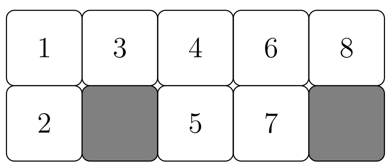
The scheduling problem we considered in this talk is defined as follows: a set $\mathcal{T} = \{1, 2, \ldots, n\}$ of n tasks is to be executed on an unlimited number of machines. Each machine can process at most one task at a time and each task is processed once. Tasks have a unit processing time and are partially ordered by a precedence graph $\mathcal{G} = (\mathcal{T}, \mathcal{A})$. Let t_i be the starting time of the task i. For any arc $(i, j) \in \mathcal{A}$, the task i must finish its execution before the task j starts executing, *i.e.* $t_i + 1 \leq t_j$. If tasks i and j are assigned to different processors, a unit communication delay must be added after the execution of the task i, to send data to task j. Thus $t_i + 2 \leq t_j$. The problem is to find a feasible schedule that minimizes the makespan; this problem is referred to $\overline{P}|prec, p_i = 1, c_{ij} = 1|C_{max}$. The purpose is to present the first fixed parameter tractable algorithm for this problem. The parameter is the maximum number of tasks that are schedulable at the same time considering task execution time windows.

Let us consider for example the precedence graph of 8 unit computing time tasks presented by Figure 1a. Figure 1b presents an associated feasible schedule of makespan 5. Associated starting times are respectively $t^\sigma = (0, 0, 1, 2, 2, 3, 3, 4)$.

Few fixed parameter algorithm are developed for scheduling problems. Mnich and van Bevern [3] surveyed main results on parameterized complexity for scheduling problems and identified 15 open problems. The scheduling problem $\overline{P}|prec, p_i = 1, c_{ij} = 1|C_{max}$ was proved to be NP-hard by Hoogeveen et al. [2]. It was intensively studied since the 1990s due to the importance of applications, *see.* the survey [1]. An exact dynamic programming algorithm of time complexity $\mathcal{O}(2^{w(G)} n^{2w(G)})$ was developed by Veltman [5]. The parameter $w(G)$ is the width of the precedence graph G. We can observe that it is not a fixed-parameter algorithm. Our algorithm is inspired from the work of Munier-Kordon [4] which developed a fixed-parameter algorithm for the problem $P|prec, p_i = 1|C_{max}$.



(a) A precedence graph $\mathcal{G} = (\mathcal{T}, \mathcal{A})$.



(b) An optimum schedule for the precedence graph $\mathcal{G} = (\mathcal{T}, \mathcal{A})$ of Figure1a.

FIG. 1: A precedence graph $\mathcal{G} = (\mathcal{T}, \mathcal{A})$ and an associated optimum schedule.

2 Description of the fixed parameter algorithm and perspectives

Let us consider a precedence graph $\mathcal{G} = (\mathcal{T}, \mathcal{A})$ and an upper bound $\overline{C}$ of the makespan. Release times r_i and deadlines d_i can be computed for every task i from the precedence graph $\mathcal{G}$. To find the optimal schedule, we build an associated multistage graph $S(\mathcal{G}) = (N, A)$ with $\overline{C}$ stages. To narrow the search space, we developed some dominant properties of optimal schedules, which can significant reduce the size of multistage graphs.

The nodes of the graph $S(\mathcal{G})$ are partitioned into $\overline{C}$ stages. For any $\alpha \in \{0, \dots, \overline{C} - 1\}$, N_α is the set of nodes at stage α. A node $p \in N$ is a couple $(W(p), B(p))$, where $B(p)$ and $W(p)$ are two subsets of tasks such that $B(p) \subseteq W(p) \subseteq \mathcal{T}$. If $p \in N_\alpha$, tasks from $W(p)$ have to be completed at time $\alpha + 1$, while those from $B(p)$ are scheduled at time α.

For any $\alpha \in \{0, 1, \dots, \overline{C} - 2\}$ and $(p, q) \in N_\alpha \times N_{\alpha+1}$, the arc $(p, q) \in A$ if there exists a feasible dominant schedule such that tasks from $W(q)$ are all completed at time $\alpha + 2$ with tasks from $B(q)$ executed at time $\alpha + 1$ and those from $B(p)$ at time α.

We showed that there exists a feasible schedule of length less than $\overline{C}$ if and only if there exists a path in $S(\mathcal{G})$ from a node $p \in N_0$ to a node q with $W(q) = \mathcal{T}$.

Figure 2 is the multistage graph associated with the precedence graph of Figure 1a and $\overline{C} = 6$. We observe that the path $(p_0, p_1^1, p_2^1, p_3^1, p_4^1)$ corresponds to the schedule shown in Figure 1b.

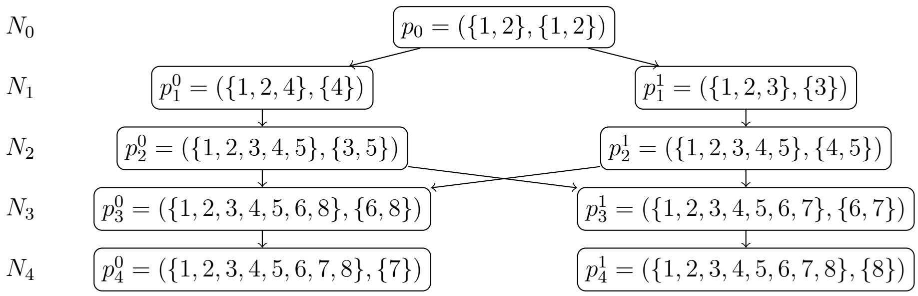


FIG. 2: The multistage graph $S(\mathcal{G})$ associated with the precedence graph of Figure 1a and $\overline{C} = 6$.

The following theorem states that this scheduling problem is fixed parameter tractable.

Theorem 1 (Complexity of the Algorithm) *The time complexity of the construction of the graph $S(\mathcal{G})$ is $\mathcal{O}(n^3 \times pw(\overline{C}) \times 2^{4pw(\overline{C})})$, where $pw(\overline{C})$ is the maximum number of tasks that are schedulable at a same time considering task execution time windows $[r_i, d_i]$, $i \in \mathcal{T}$.*

This is the first fixed-parameter algorithm for this scheduling problem. In the future, we plan to extend this approach to other criteria like L_{max} and to typed tasks system models.

References

[1] Maciej Drozdowski. *Scheduling for Parallel Processing*. Springer, 2009.

[2] J.A. Hoogeveen, J.K. Lenstra, and B. Veltman. *Three, four, five, six, or the complexity of scheduling with communication delays*. Vol. 16. 3. 1994, pp. 129–137.

[3] Matthias Mnich and René Van Bevern. “Parameterized complexity of machine scheduling: 15 open problems”. In: *Computers Operations Research* 100 (2018).

[4] Alix Munier Kordon. “A fixed-parameter algorithm for scheduling unit dependent tasks on parallel machines with time windows”. In: *Discrete Applied Mathematics* 290 (2021).

[5] Bart Veltman. “Multiprocessor scheduling with communication delays”. PhD thesis. Eindhoven University of Technology, 1993.