



**HAL**  
open science

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

Ning Tang, Alix Munier Kordon

► **To cite this version:**

Ning Tang, Alix Munier Kordon. A Fixed-Parameter Algorithm for Scheduling Unit dependent Tasks with Unit Communication Delays. Euro-Par 2021 - 27th International European Conference on Parallel and Distributed Computing, Aug 2021, Lisbon, Portugal. pp.105-119, 10.1007/978-3-030-85665-6\_7. hal-03816158

**HAL Id: hal-03816158**

**<https://hal.science/hal-03816158v1>**

Submitted on 15 Oct 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Ning Tang<sup>[0000–0002–1388–8788]</sup> and Alix Munier Kordon<sup>[0000–0002–2170–6366]</sup>

Sorbonne Université, CNRS, LIP6, F-75005 Paris, France  
Ning.Tang@lip6.fr, Alix.Munier@lip6.fr

**Abstract.** This paper considers the minimization of the makespan for a set of dependent tasks with unit duration and unit communication delays. Given an upper bound of the makespan, release dates and deadlines of the tasks can be computed. Time windows are defined accordingly. We prove that our scheduling problem is fixed-parameter tractable; the parameter is the maximum number of tasks that are schedulable at the same time considering time windows.

A fixed-parameter algorithm based on a dynamic programming approach is developed and proved to solve this optimization problem. This is, as far as we know, the first fixed-parameter algorithm for a scheduling problem with communication delays.

**Keywords:** scheduling · communication delays · makespan · fixed-parameter algorithm.

## 1 Introduction

This paper tackles a basic scheduling problem with communication delays defined as follows: a set  $\mathcal{T} = \{1, 2, \dots, n\}$  of  $n$  tasks is to be executed on an unlimited number of machines (sometimes also called as processors). Each machine can process at most one task at a time and each task is processed once. Tasks have a unit execution 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  $j$  and thus  $t_i + 2 \leq t_j$ . The problem is to find a feasible schedule that minimizes the makespan; it is referred to  $\overline{P}|prec, p_i = 1, c_{ij} = 1|C_{max}$  using standard notations [12].

The development of fixed-parameter algorithms for NP-complete problems is a way to get polynomial-time algorithms when some parameters are fixed [7, 9]. More formally, a fixed-parameter algorithm solves any instance of a problem of size  $n$  in time  $f(k) \cdot \text{poly}(n)$ , where  $f$  is allowed to be a computable superpolynomial function and  $k$  the associated parameter.

Mnich and van Bevern [14] surveyed main results on parameterized complexity for scheduling problems and identified 15 open problems. However, there is no result of parameterized complexity for scheduling problems with communication delays.

The purpose of this paper is to present the first fixed-parameter algorithm for the problem  $\overline{P}|prec, p_i = 1, c_{ij} = 1|C_{max}$ . We observe that, to any upper bound  $\overline{C}$  of the minimum makespan, feasible release dates  $r_i$  and deadlines  $d_i$  can be associated for any task  $i \in \mathcal{T}$  considering the precedence graph  $\mathcal{G}$ .

The parameter considered for our algorithm is the pathwidth, denoted by  $pw(\overline{C})$ , and corresponds to the maximum number of tasks minus one that can be executed simultaneously if we only consider the intervals  $\{(r_i, d_i), i \in \mathcal{T}\}$ .

The pathwidth  $pw(\overline{C})$  can be interpreted as a simple measure of the parallelism of the instance considered for a fixed makespan  $\overline{C}$ . One can observe that  $pw(\overline{C})$  is the pathwidth of the interval graph associated with the set of intervals  $\{(r_i, d_i), i \in \mathcal{T}\}$  [4].

We prove in this paper that the scheduling problem with communication delays  $\overline{P}|prec, p_i = 1, c_{ij} = 1|C_{max}$  can be solved in time  $\mathcal{O}(n^3 \cdot pw(\overline{C}) \cdot 2^{4pw(\overline{C})})$  using a dynamic programming approach. A multistage graph where paths model feasible schedules is partially built until a complete feasible schedule is obtained. Our algorithm is inspired from the work of Munier [15] which developed a fixed-parameter algorithm for the problem  $P|prec, p_i = 1|C_{max}$ .

This paper is organised as follows. Section 2 presents related work. Section 3 defines the problem and the notations. It also recalls the modeling of our problem using an integer linear program. Section 4 presents some important dominance properties considered to characterize the structure of the solutions. Section 5 is dedicated to the description of the algorithm and its validity proof. The complexity of our algorithm is studied in Section 6. Section 7 is our conclusion.

## 2 Related work

The scheduling problem  $P|prec, p_i = 1, c_{ij} = 1|C_{max}$  with a limited number of processors was first introduced by Rayward-Smith [18]. Basic scheduling problems with communication delays were intensively studied since the 1990s due to the importance of applications, see. the surveys [6, 11, 21, 10].

Hoogeveen et al. [13] have shown that a polynomial-time algorithm without duplication exists for solving the problem  $\overline{P}|prec, p_i = 1, c_{ij} = 1|C_{max}$  when the makespan is bounded by 5, but it is NP-complete when the makespan is bounded by 6. This problem was also proved to be polynomial-time solvable for some special classes of graphs such as trees [5], series-parallel graphs [16] and generalized n-dimensional grid task graphs [3].

Many authors considered scheduling problems with communication delays for a limited number of processors. An exact dynamic programming algorithm of time complexity  $\mathcal{O}(2^{w(G)} \cdot n^{2w(G)})$  was developed by Veltman [22] for  $P|prec, p_i = 1, c_{ij} = 1|C_{max}$ . The parameter  $w(G)$  is the width of the precedence graph  $G$

defined as the size of its largest antichain. This algorithm can clearly be considered for solving the problem without limitation of the number of machines by setting the number of machines equal to the number of tasks. We can observe that it is not a fixed-parameter algorithm. Zinder et al. [24] have developed an exact branch-and-bound algorithm which converges to an optimal schedule for the problem  $P|prec, p_i = 1, c_{ij} = 1|C_{max}$ . For the more general problem  $P|prec, c_{ij}|C_{max}$ , Sinnens et al. in [20] have developed an enumerative  $A^*$  algorithm coupled with pruning methods. Orr and Sinnens [17] have developed an original techniques to reduce the space of exploration and to speed up branch-and-bound methods.

Several authors also considered integer linear programming formulations (ILP in short) to solve exactly scheduling problems with communications delays and a limited number of processors. Davidović et al. in [8] tackled the scheduling problems for a fixed network of processors; communications are proportional to both the amount of exchanged data between pairs of dependent tasks and the distance between processors in the multiprocessor architecture. They developed two formulations and they compared them experimentally. Later, Ait El Cadi et al. [2] improved this approach by reducing the size of the linear program (number of variables and constraints) and by adding cuts; they compared positively to the previous authors. Venugopalan and Sinnens in [23] provided a new ILP formulation for the usual problem  $P|prec, c_{ij}|C_{max}$  and comparison with [8] for several classes of graphs and fixed number of processors.

Extensions of usual problems with communication delays were extensively studied. For example, the survey of Giroudeau and Koenig [11] considered a hierarchical communication model where processors are grouped into clusters. Shimada et al. [19] developed two heuristic based methods to consider both malleable tasks and communications delays for executing a program on an homogeneous multi-core computing system. Ait-Aba et al. [1] provided complexity results for an extension of the basic communication model for scheduling problems on an heterogeneous computing systems with two different resources.

### 3 Problem definition and notations

In this section, we first recall that our scheduling problem can be modelled using an integer linear program. The computation of release dates and deadlines of tasks are expressed depending on the precedence graph and a fixed upper bound of minimum makespan  $\bar{C}$ . The associated pathwidth is defined, and a necessary condition for feasible schedules is provided. They will be both considered in next sections for our fixed-parameter algorithm.

#### 3.1 Problem definition

An instance of our scheduling problem  $\bar{P}|prec, p_i = 1, c_{ij} = 1|C_{max}$  is defined by a directed acyclic graph  $\mathcal{G} = (\mathcal{T}, \mathcal{A})$ . For each task  $i \in \mathcal{T}$ , let  $\Gamma^+(i)$  (resp.  $\Gamma^-(i)$ )

be the set of successors (resp. predecessors) of  $i$ , i.e.  $\Gamma^+(i) = \{j \in \mathcal{T}, (i, j) \in \mathcal{A}\}$  and  $\Gamma^-(i) = \{j \in \mathcal{T}, (j, i) \in \mathcal{A}\}$ .

Our scheduling problem can easily be modelled by a integer linear program  $P$  defined below. For any task  $i \in \mathcal{T}$ , we note  $t_i$  the starting time of the execution of task  $i$ . For any arc  $e = (i, j) \in \mathcal{A}$ , we note  $x_{ij}$  the communication delay between the tasks  $i$  and  $j$ . We set  $x_{ij} = 0$  if the task  $j$  is executed just after the task  $i$  on the same processor; in this case, there is no communication delay between them. Otherwise,  $x_{ij} = 1$ .

$$(P) \begin{cases} \min C \\ \forall e = (i, j) \in \mathcal{A}, t_i + 1 + x_{ij} \leq t_j & (1) \\ \forall i \in \mathcal{T}, t_i + 1 \leq C & (2) \\ \forall i \in \mathcal{T}, \sum_{j \in \Gamma^+(i)} x_{ij} \geq |\Gamma^+(i)| - 1 & (3) \\ \forall i \in \mathcal{T}, \sum_{j \in \Gamma^-(i)} x_{ji} \geq |\Gamma^-(i)| - 1 & (4) \\ \forall i \in \mathcal{T}, t_i \in \mathbb{N} \\ \forall e = (i, j) \in \mathcal{A}, x_{ij} \in \{0, 1\} \end{cases}$$

Variables are the starting times  $t_i, \forall i \in \mathcal{T}$  of the tasks, the communication delays  $x_{ij}, \forall (i, j) \in \mathcal{A}$  and the makespan  $C$ . Inequalities (1) express precedence relations and communication delays between tasks executions. Inequalities (2) define the makespan. Inequalities (3) express that any task has at most one successor performed at its completion time on the same processor. Similarly, inequalities (4) express that any task has at most one predecessor performed just before its starting time on the same processor.

Any feasible schedule  $\sigma(\mathcal{G})$  corresponds to a feasible solution of  $P$  and is thus defined by two vectors: starting times  $t^\sigma \in \mathbb{N}^{|\mathcal{T}|}$  and communication delays  $x^\sigma \in \{0, 1\}^{|\mathcal{A}|}$ .

Now, one can observe that  $x$  is the decision variable of  $P$ , and thus from a practical point of view, we can consider the starting time  $t \in (\mathbb{R}^+)^n$ . Anyway, we will limit in this paper starting times to non negative integer values.

Let us consider for example the precedence graph presented by Figure 1a for  $\mathcal{T} = \{1, 2, \dots, 8\}$ . Figure 1b presents an associated feasible schedule  $\sigma(\mathcal{G})$  of makespan 5. Associated starting times and communication delays are respectively  $t^\sigma = (0, 0, 1, 2, 2, 3, 3, 4)$  and  $x^\sigma = (x_{13}^\sigma, x_{14}^\sigma, x_{15}^\sigma, x_{25}^\sigma, x_{36}^\sigma, x_{46}^\sigma, x_{57}^\sigma, x_{58}^\sigma) = (0, 1, 1, 0, 1, 0, 0, 1)$ .

### 3.2 A necessary condition on feasible schedules

A time window  $(r_i, d_i)$  associated to the task  $i \in \mathcal{T}$  is given by a release date  $r_i \in \mathbb{N}$  and a deadline  $d_i \in \mathbb{N}$  such that the task  $i$  must be completed during the time interval  $(r_i, d_i)$ , i.e.  $t_i \geq r_i$  and  $t_i + 1 \leq d_i$ .

Let us suppose that the predecessors  $j_1, \dots, j_p$  of any task  $i \in \mathcal{T}$  are numbered in decreasing order of the release times, i.e.  $r_{j_1} \geq r_{j_2} \geq \dots \geq r_{j_p}$ . Then,

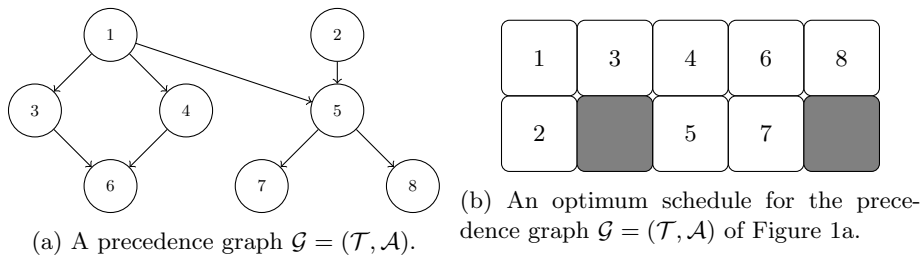


Fig. 1: A precedence graph  $\mathcal{G} = (\mathcal{T}, \mathcal{A})$  and an associated optimum schedule.

the release date  $r_i$  can be calculated recursively as follows:

$$r_i = \begin{cases} 0 & \text{if } |\Gamma^-(i)| = 0 \\ r_{j_1} + 1 & \text{if } |\Gamma^-(i)| = 1 \text{ or } (|\Gamma^-(i)| > 1 \text{ and } r_{j_1} > r_{j_2}) \\ r_{j_1} + 2 & \text{if } |\Gamma^-(i)| > 1 \text{ and } r_{j_1} = r_{j_2}. \end{cases} \quad (1)$$

Now, let  $\bar{C}$  be an upper bound of the minimum makespan for the graph  $\mathcal{G} = (\mathcal{T}, \mathcal{A})$ . Deadlines of tasks can similarly be computed as follows. Let us suppose that the successors  $k_1, k_2, \dots, k_q$  of any task  $i \in \mathcal{T}$  are numbered in the increasing order of the deadlines, i.e.  $d_{k_1} \leq d_{k_2} \leq \dots \leq d_{k_q}$ . The deadline  $d_i$  can be calculated by the following recursive function:

$$d_i = \begin{cases} \bar{C} & \text{if } |\Gamma^+(i)| = 0 \\ d_{k_1} - 1 & \text{if } |\Gamma^+(i)| = 1 \text{ or } (|\Gamma^+(i)| > 1 \text{ and } d_{k_1} < d_{k_2}) \\ d_{k_1} - 2 & \text{if } |\Gamma^+(i)| > 1 \text{ and } d_{k_1} = d_{k_2}. \end{cases} \quad (2)$$

Clearly, for any schedule  $\sigma(\mathcal{G})$  of makespan bounded by  $\bar{C}$ ,  $t_i^\sigma \in [r_i, d_i]$ . For any value  $\alpha \in \{0, \dots, \bar{C}-1\}$ , we note  $X_\alpha$  as the set of tasks that can be scheduled at time  $\alpha$  following release times and deadlines, i.e.  $X_\alpha = \{i \in \mathcal{T}, r_i \leq \alpha, \alpha+1 \leq d_i\}$ . We also denote by  $Z_\alpha$  the set of tasks that must be completed at or before time  $\alpha+1$ , i.e.  $Z_\alpha = \{i \in \mathcal{T}, d_i \leq \alpha+1\}$ . The pathwidth of  $\mathcal{G}$  associated to the length  $\bar{C}$  is defined as  $pw(\bar{C}) = \max_{\alpha \in \{0, \dots, \bar{C}-1\}} (|X_\alpha| - 1)$ .

Figure 2a shows the release time and deadline of tasks from the graph presented by Figure 1a with the upper bound of the makespan  $\bar{C} = 6$ . Figure 2b shows the associated sets  $X_\alpha$  and  $Z_\alpha$  for  $\alpha \in \{0, 1, \dots, 5\}$ . For this example,  $pw(\bar{C}) = |X_3| - 1 = 5$ .

Let us consider that  $\sigma(\mathcal{G})$  is a feasible schedule of makespan  $C \leq \bar{C}$ . For every integer  $\alpha \in \{0, \dots, C-1\}$ , we set  $\mathcal{T}_\alpha^\sigma = \{i \in \mathcal{T}, t_i^\sigma = \alpha\}$ . The following lemma will be considered further to reduce the size of the tasks sets built at each step of our algorithm.

**Lemma 1.** *Let  $\sigma(\mathcal{G})$  be a feasible schedule of  $\mathcal{G}$ . For any  $\alpha \in \{0, \dots, C-1\}$ ,*

$$\bigcup_{\beta=0}^{\alpha} \mathcal{T}_\beta^\sigma - Z_\alpha \subseteq X_\alpha \cap X_{\alpha+1}.$$

tasks	1	2	3	4	5	6	7	8
$r_i$	0	0	1	1	2	3	3	3
$d_i$	3	3	5	5	4	6	6	6

(a) Release times  $r_i$  and deadlines  $d_i$  of all tasks from Figure 1a for  $\bar{C} = 6$ .

$\alpha$	$X_\alpha$	$Z_\alpha$
0	{1, 2}	$\emptyset$
1	{1, 2, 3, 4}	$\emptyset$
2	{1, 2, 3, 4, 5}	{1, 2}
3	{3, 4, 5, 6, 7, 8}	{1, 2, 5}
4	{3, 4, 6, 7, 8}	{1, 2, 3, 4, 5}
5	{6, 7, 8}	{1, 2, 3, 4, 5, 6, 7, 8}

(b) Sets  $X_\alpha$  and  $Z_\alpha$ ,  $\alpha \in \{0, \dots, 5\}$ .Fig. 2: Release times, deadlines, and sets  $X_\alpha$  and  $Z_\alpha$ ,  $\alpha \in \{0, \dots, 5\}$  for the instance presented by Figure 1a and  $\bar{C} = 6$ .

*Proof.* Since  $\sigma(\mathcal{G})$  is feasible, for any  $\alpha \in \{0, \dots, C-1\}$ ,  $\mathcal{T}_\alpha^\sigma \subseteq X_\alpha$  and thus,  $\forall i \in \bigcup_{\beta=0}^{\alpha} \mathcal{T}_\beta^\sigma$ ,  $r_i \leq \alpha$ . Moreover, each task  $i \notin Z_\alpha$  satisfies  $d_i \geq \alpha + 2$ .

Thus, for any task  $i \in \bigcup_{\beta=0}^{\alpha} \mathcal{T}_\beta^\sigma - Z_\alpha$ ,  $[\alpha, \alpha + 2] \subseteq [r_i, d_i]$ . Therefore  $\bigcup_{\beta=0}^{\alpha} \mathcal{T}_\beta^\sigma - Z_\alpha \subseteq X_\alpha \cap X_{\alpha+1}$ , and the lemma is proved.  $\square$

For our example presented by Figure 2 and  $\alpha = 3$ , we have  $Z_3 = \{1, 2, 5\}$  and  $X_3 \cap X_4 = \{3, 4, 6, 7, 8\}$ . For the schedule showed in Figure 1, we have  $\bigcup_{\beta=0}^3 \mathcal{T}_\beta^\sigma = \{1, 2, 3, 4, 5, 6, 7\}$ . We observe that  $\bigcup_{\beta=0}^3 \mathcal{T}_\beta^\sigma - Z_3 = \{3, 4, 6, 7\} \subseteq X_3 \cap X_4$ .

## 4 Dominance properties

In this section, we express two properties of optimal schedules, which can narrow the search space.

### 4.1 Coherent schedules

We can associate to any execution time vector  $t^\sigma \in \mathbb{N}^n$  of a feasible schedule a communication vector  $x^\sigma$ , where each element  $x_i^\sigma \in \{0, 1\}$  in  $x^\sigma$  is of maximum value. The schedule obtained is said to be coherent, as defined as follows:

**Definition 1 (Coherent schedule).** A schedule  $\sigma(\mathcal{G})$  is **coherent** if for any arc  $e = (i, j) \in \mathcal{A}$ , if  $t_i^\sigma + 2 \leq t_j^\sigma$ , then  $x_{ij}^\sigma = 1$ .

Next Lemma is a consequence of the ILP formulation of our scheduling problem:

**Lemma 2 (Coherent schedule property).** Let  $\sigma(\mathcal{G})$  be a feasible schedule of  $\mathcal{G}$ . Then, there exists a coherent feasible schedule  $\sigma'(\mathcal{G})$  with  $t_i^{\sigma'} = t_i^\sigma, \forall i \in \mathcal{T}$ .

*Proof.* Let us suppose that  $\sigma(\mathcal{G})$  is a non-coherent feasible schedule. Then, we build a schedule  $\sigma'(\mathcal{G})$  by setting:

1. for every task  $i \in \mathcal{T}$ ,  $t_i^{\sigma'} = t_i^\sigma$ ;
2. for every arc  $(i, j) \in \mathcal{A}$ , if  $t_i^{\sigma'} + 2 > t_j^{\sigma'}$ , then  $x_{ij}^{\sigma'} = 0$ , otherwise,  $x_{ij}^{\sigma'} = 1$ .

We show that  $\sigma'(\mathcal{G})$  is feasible. Let us consider an arc  $e = (i, j) \in \mathcal{A}$ . Two cases are considered:

1. if  $t_i^\sigma + 2 > t_j^\sigma$ , then  $x_{ij}^\sigma = 0 = x_{ij}^{\sigma'}$  and  $t_i^{\sigma'} + 1 \leq t_j^{\sigma'}$ , the constraint is thus fulfilled by  $\sigma'(\mathcal{G})$ .
2. Now, if  $t_i^\sigma + 2 \leq t_j^\sigma$ , we get  $x_{ij}^{\sigma'} = 1 \geq x_{ij}^\sigma$  and  $t_i^{\sigma'} + 1 + x_{ij}^{\sigma'} \leq t_j^{\sigma'}$  is true.

Moreover, for every arc  $(i, j) \in \mathcal{A}$ ,  $x_{ij}^{\sigma'} \geq x_{ij}^\sigma$ . Thus  $\sigma'(\mathcal{G})$  is a coherent feasible schedule and we proved the lemma.  $\square$

## 4.2 Preferred sons

Let us consider that  $\sigma(\mathcal{G})$  is a feasible coherent schedule of makespan  $C \leq \bar{C}$ . For every integer  $\alpha \in \{0, \dots, C-1\}$ , we set  $W_\alpha = \bigcup_{\beta=0}^{\alpha} \mathcal{T}_\beta^\sigma$  and  $B_\alpha = \mathcal{T}_\alpha^\sigma$ . The set  $W_\alpha$  contains all the tasks that are performed during the interval  $[0, \alpha + 1]$ , and  $B_\alpha$  contains all the tasks that are executed at time  $\alpha$ . We show hereafter that, if a task  $i \in \mathcal{T}$  has one or more successors schedulable at time  $t_i^\sigma + 1$  (called the preferred sons of  $i$ ), then we can always impose that exactly one of them is executed at time  $t_i^\sigma + 1$  on the same processor as the task  $i$ .

**Definition 2 (Preferred sons).** For every integer  $\alpha \in \{0, \dots, C-1\}$ , a task  $j \in \mathcal{T}$  is a **preferred son** of a task  $i \in B_\alpha$  if  $j$  is a successor of  $i$  that is schedulable at time  $\alpha + 1$ . The set of the **preferred sons** of  $i$  with respect to  $W_\alpha$  and  $B_\alpha$  is defined as  $PS_{W_\alpha, B_\alpha}(i) = \{j \in \Gamma^+(i), \Gamma^-(j) \subseteq W_\alpha \text{ and } \Gamma^-(j) \cap B_\alpha = \{i\}\}$ .

**Definition 3 (Preferred sons property).** A coherent feasible schedule  $\sigma(\mathcal{G})$  satisfies the **preferred sons property** if, for every integer  $\alpha \in \{0, \dots, C-1\}$ , each task  $i \in B_\alpha$  such that  $PS_{W_\alpha, B_\alpha}(i) \neq \emptyset$  has exactly one preferred son executed at time  $\alpha + 1$ .

Let us consider as example the precedence graph and the feasible schedule presented by Figure 1. For  $\alpha = 0$ ,  $W_0 = B_0 = \{1, 2\}$ ,  $PS_{W_0, B_0}(1) = \{3, 4\}$  and  $PS_{W_0, B_0}(2) = \emptyset$ . Thus, we can enforce that exactly one task in  $\{3, 4\}$  would be executed at time 1.

**Lemma 3.** Let  $\sigma(\mathcal{G})$  be a coherent feasible schedule. There exists a corresponding coherent feasible schedule  $\sigma'(\mathcal{G})$  that satisfies the preferred sons property and such that for any task  $i \in \mathcal{T}$ ,  $t_i^{\sigma'} \leq t_i^\sigma$ .

*Proof.* We can suppose without loss of generality that tasks are scheduled by  $\sigma(\mathcal{G})$  as soon as possible following the communication delay vector  $x^\sigma$  of  $\sigma(\mathcal{G})$ , i.e.  $\forall i \in \mathcal{T}$ ,  $t_i^\sigma = \max(0, \max_{j \in \Gamma^-(i)}(t_j^\sigma + 1 + x_{ji}^\sigma))$ .



Let us suppose that  $\sigma(\mathcal{G})$  does not verify the preferred sons property. Let then  $\alpha \in \{0, \dots, C-1\}$  be the first instant for which the property is not fulfilled, and  $i^* \in B_\alpha$  a corresponding task with  $PS_{W_\alpha, B_\alpha}(i^*) \neq \emptyset$ . We show that, for every task  $j \in \Gamma^+(i^*)$ ,  $x_{i^*j}^\sigma = 1$ .

- Since  $i^*$  is performed at time  $\alpha$ ,  $i^*$  cannot have two successors scheduled at time  $\alpha + 1$ . So, every task  $j \in PS_{W_\alpha, B_\alpha}(i^*)$  satisfies  $t_j^\sigma \geq t_{i^*}^\sigma + 2$  and by coherence of  $\sigma(\mathcal{G})$ ,  $x_{i^*j}^\sigma = 1$ .
- Now, any task  $j \in \Gamma^+(i^*) - PS_{W_\alpha, B_\alpha}(i^*)$  is not schedulable at time  $\alpha + 1$ , thus  $t_j^\sigma \geq t_{i^*}^\sigma + 2$  and by coherence of  $\sigma(\mathcal{G})$ ,  $x_{i^*j}^\sigma = 1$ .

Now, any task  $j \in PS_{W_\alpha, B_\alpha}(i^*)$  has all its predecessors in  $W_\alpha$ , thus  $\forall k \in \Gamma^-(j)$ ,  $t_k^\sigma + 2 \leq t_{i^*}^\sigma + 2 \leq t_j^\sigma$ , and by coherence of  $\sigma(\mathcal{G})$ ,  $x_{kj}^\sigma = 1$ . We build another coherent schedule  $\sigma'(\mathcal{G})$  as follows:

1. We first choose a task  $j^* \in PS_{W_\alpha, B_\alpha}(i^*)$ . We then set  $x_{i^*j^*}^{\sigma'} = 0$  and for each arc  $e = (k, \ell) \in \mathcal{A} - \{(i^*, j^*)\}$ ,  $x_{k\ell}^{\sigma'} = x_{k\ell}^\sigma$ .
2. We set  $\forall i \in \mathcal{T}$ ,  $t_i^{\sigma'} = \max(0, \max_{j \in \Gamma^-(i)}(t_j^{\sigma'} + 1 + x_{ji}^{\sigma'}))$ .

For every task  $i \in \mathcal{T}$ ,  $t_i^{\sigma'} \leq t_i^\sigma$ . Now, we get  $\sum_{\ell \in \Gamma^+(i^*)} x_{i^*\ell}^{\sigma'} = \sum_{\ell \in \Gamma^+(i^*) - \{j^*\}} x_{i^*\ell}^{\sigma'} + x_{i^*j^*}^{\sigma'} = |\Gamma^+(i^*)| - 1$ . Similarly, we get  $\sum_{\ell \in \Gamma^-(j^*)} x_{\ell j^*}^{\sigma'} = \sum_{\ell \in \Gamma^-(j^*) - \{i^*\}} x_{\ell j^*}^{\sigma'} + x_{i^*j^*}^{\sigma'} = |\Gamma^-(j^*)| - 1$ , and thus  $x^{\sigma'}$  is feasible.

Each task  $i^*$  is considered at most once and thus this transformation is done at most  $n$  times. So, it gives a feasible coherent schedule that satisfies the preferred sons property without increasing the makespan, thus the lemma holds.  $\square$

### 4.3 Limitation of the feasible schedules set

The following theorem is a simple consequence of Lemma 2 and 3.

**Theorem 1.** *Let  $\sigma(\mathcal{G})$  be a feasible schedule. Then, there exists a feasible coherent  $\sigma'(\mathcal{G})$  that satisfies the preferred sons property and such that, for each task  $i \in \mathcal{T}$ ,  $t_i^{\sigma'} \leq t_i^\sigma$ .*

## 5 Presentation of the algorithm

This section provides our fixed-parameter algorithm. We start with the description of the multistage graph. We present then our algorithm and we show its correctness.

### 5.1 Description of the multistage graph

Let us consider a precedence graph  $\mathcal{G} = (\mathcal{T}, \mathcal{A})$  and an upper bound  $\bar{C}$  of the makespan. We build an associated multistage graph  $S(\mathcal{G}) = (N, A)$  with  $\bar{C}$  stages which maximum paths represent all the feasible schedules following the condition of Theorem 1.

*Nodes of  $S(\mathcal{G})$*  Elements of  $N$  are partitioned into  $\overline{C}$  stages. For any value  $\alpha \in \{0, \dots, \overline{C} - 1\}$ ,  $N_\alpha$  is the set of nodes at stage  $\alpha$ . A node  $p \in N$  is a couple  $(W(p), B(p))$ , where  $B(p) \subseteq W(p) \subseteq \mathcal{T}$ . 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  $\alpha$ .  $N_0$  contains only one node  $p_0$  with  $B(p_0) = \{i \in \mathcal{T}, \Gamma^-(i) = \emptyset\}$  and  $W(p_0) = B(p_0)$ .

Observe that, for any value  $\alpha \in \{0, \dots, \overline{C} - 1\}$ , all tasks from  $Z_\alpha$  must be completed at time  $\alpha + 1$ , thus for any node  $p \in N_\alpha$ ,  $Z_\alpha \subseteq W(p)$ . Moreover, by Lemma 1,  $W(p) - Z_\alpha \subseteq X_\alpha \cap X_{\alpha+1}$ .

*Arcs of  $S(\mathcal{G})$*  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 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  $\alpha$ . The nodes  $p$  and  $q$  satisfy then the following conditions:

- A.1 Since  $p$  is associated to a partial schedule of  $q$ ,  $W(p) \cup B(q) = W(q)$  and since tasks can only be executed once,  $W(p) \cap B(q) = \emptyset$ .
- A.2 Any task  $i \in B(q)$  must be schedulable at time  $\alpha + 1$ , thus all its predecessors must belong to  $W(p)$ . Then,  $B(q) \subseteq \{i \in X_{\alpha+1}, \Gamma^-(i) \subseteq W(p)\}$ .
- A.3 Any task  $i \in B(q)$  cannot have more than one predecessor scheduled at time  $\alpha$ , thus  $B(q) \subseteq \{i \in X_{\alpha+1}, |\Gamma^-(i) \cap B(p)| \leq 1\}$ .
- A.4 Any task  $i \in X_{\alpha+1} - W(p)$  for which all its predecessors are completed at time  $\alpha$  must be scheduled at time  $\alpha + 1$ . Thus, if  $\Gamma^-(i) \subseteq W(p) - B(p)$ , then  $i \in B(q)$ .
- A.5 For any task  $i \in B(p)$ , if  $PS_{W(p), B(p)}(i) \cap X_{\alpha+1} \neq \emptyset$ , then by Definition 2, these successors of  $i$  are schedulable at time  $\alpha + 1$ . Following Theorem 1, we impose that exactly one among them is executed at time  $\alpha$  on the same processor than  $i$  and thus  $|PS_{W(p), B(p)}(i) \cap B(q)| = |\Gamma^+(i) \cap B(q)| = 1$ . Otherwise, if  $PS_{W(p), B(p)}(i) \cap X_{\alpha+1} = \emptyset$ , no successor of  $i$  can be scheduled at time  $\alpha + 1$  which corresponds to  $|\Gamma^+(i) \cap B(q)| = 0$ .

*Remark 1.* The preferred sons of a task  $i \in \mathcal{T}$  were initially defined with respect to two sets of tasks  $W_\alpha$  and  $B_\alpha$  built from a feasible schedule. Here, for any node  $p \in N_\alpha$ , the definition of  $PS$  is extended to consider the sets  $W(p)$  and  $B(p)$  simply by assuming that tasks from  $W(p)$  (resp.  $B(p)$ ) are those which are completed at time  $\alpha + 1$  (resp. performed at time  $\alpha$ ).

Figure 3 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. On the same way, the path  $(p_0, p_1^0, p_2^0, p_3^0, p_4^0)$  corresponds to the schedule shown by Figure 4.

## 5.2 Description of the algorithm

Algorithm 1 builds iteratively the multistage graph  $S(\mathcal{G}) = (N, A)$ . This algorithm returns false if there is no feasible schedule of makespan bounded by  $\overline{C}$ , otherwise it returns the optimum makespan. For any set of tasks  $X \subseteq \mathcal{T}$ , let  $\mathcal{P}(X)$  be the power set of  $X$ , i.e. the set of all subsets of  $X$  including the empty

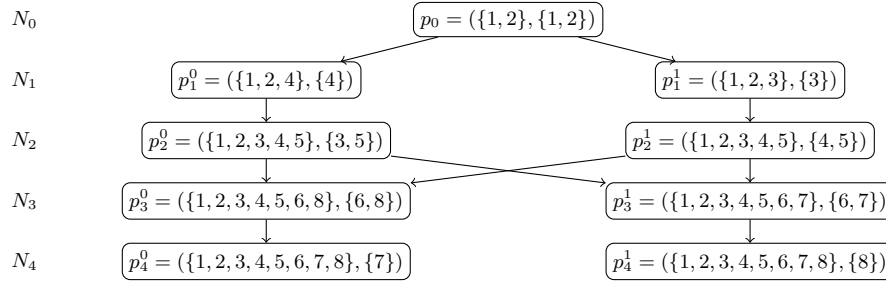


Fig. 3: The multistage graph associated with the precedence graph of Figure 1a and  $\bar{C} = 6$ .



Fig. 4: An optimum schedule corresponding to the path  $(p_0, p_1^0, p_2^0, p_3^0, p_4^0)$  of Figure 3.

ones. This algorithm is composed by three main sections. Lines 1 – 6 correspond to the initialization step. Lines 7 – 9 build all the possible nodes. Lines 10 – 17 build the arcs and delete all the non connected nodes.

### 5.3 Validity of the algorithm

**Lemma 4.** *Any feasible schedule  $\sigma(\mathcal{G})$  of makespan  $C \leq \bar{C}$  corresponds to a path of  $S(\mathcal{G})$  ending with a node  $p$  with  $W(p) = \mathcal{T}$ .*

*Proof.* Let suppose that  $\sigma(\mathcal{G})$  is a feasible schedule of makespan  $C \leq \bar{C}$ . By Theorem 1, we can suppose that  $\sigma(\mathcal{G})$  is coherent and satisfies the preferred sons property. We can also suppose that  $\sigma(\mathcal{G})$  is an as-soon-as-possible schedule, i.e. for any task  $i \in \mathcal{T}$ ,  $t_i^\sigma = \max(0, \max_{i \in \Gamma^-(j)}(t_j^\sigma + 1 + x_{ji}))$ .

Let us consider the sequence  $q_\alpha = (W(q_\alpha), B(q_\alpha))$  defined as  $W(q_\alpha) = \bigcup_{\beta=0}^\alpha \mathcal{T}_\beta^\sigma$  and  $B(q_\alpha) = \mathcal{T}_\alpha^\sigma$  for  $\alpha \in \{0, \dots, C-1\}$ .

For  $\alpha = 0$ ,  $W(q_0) = \mathcal{T}_0^\sigma = \{i \in \mathcal{T}, t_i^\sigma = 0\} = \{i \in \mathcal{T}, \Gamma^-(i) = \emptyset\} = W(p_0)$  and thus  $q_0 = p_0$ .

Since  $\sigma(\mathcal{G})$  is feasible, for every value  $\alpha \in \{0, \dots, C-1\}$ ,  $\mathcal{T}_\alpha^\sigma \subseteq X_\alpha$ . According to Lemma 1,  $\bigcup_{\beta=0}^\alpha \mathcal{T}_\beta^\sigma - Z_\alpha \subseteq X_\alpha \cap X_{\alpha+1}$ . So the node  $q_\alpha = (W(q_\alpha), B(q_\alpha))$  has been built at stage  $\alpha$ .

We prove then that, for every value  $\alpha \in \{0, \dots, C-2\}$ ,  $(q_\alpha, q_{\alpha+1}) \in A$ .

- $W(q_{\alpha+1}) = \bigcup_{\beta=0}^{\alpha+1} \mathcal{T}_\beta^\sigma = \bigcup_{\beta=0}^\alpha \mathcal{T}_\beta^\sigma \cup \mathcal{T}_{\alpha+1}^\sigma = W(q_\alpha) \cup B(q_{\alpha+1})$ . Moreover,  $W(q_\alpha) \cap B(q_{\alpha+1}) = \bigcup_{\beta=0}^\alpha \mathcal{T}_\beta^\sigma \cap \mathcal{T}_{\alpha+1}^\sigma = \emptyset$ . Thus, A.1 is verified.

---

**Algorithm 1:** Optimum makespan  $C \leq \bar{C}$  if it exists, false otherwise.

---

**Input:** A precedence graph  $\mathcal{G} = (\mathcal{T}, \mathcal{A})$ , an upper bound of the makespan  $\bar{C}$   
**Output:** Optimum makespan  $C \leq \bar{C}$  if it exists, false otherwise

- 1 **for**  $i \in \mathcal{T}$  **do**
- 2      $\lfloor$  Calculate  $r_i$  and  $d_i$
- 3 **for**  $\alpha \in \{0, 1, \dots, \bar{C} - 1\}$  **do**
- 4      $\lfloor$  Calculate  $X_\alpha$  and  $Z_\alpha$
- 5     The set of arcs  $A = \emptyset$
- 6     Let  $N_0 = \{p_0\}$  with  $B(p_0) = \{i \in \mathcal{T}, \Gamma^-(i) = \emptyset\}$  and  $W(p_0) = B(p_0)$
- 7 **for**  $\alpha \in \{1, 2, \dots, \bar{C} - 1\}$  **do**
- 8      $\lfloor$  Build the sets  $\mathcal{P}(X_\alpha \cap X_{\alpha+1})$  and  $\mathcal{P}(X_\alpha)$
- 9      $\lfloor$   $N_\alpha = \{p = (W, B), W = Y \cup Z_\alpha, Y \in \mathcal{P}(X_\alpha \cap X_{\alpha+1}), B \in \mathcal{P}(X_\alpha), B \subseteq W\}$
- 10 **for**  $\alpha \in \{0, 2, \dots, \bar{C} - 2\}$  **do**
- 11      $\lfloor$  **for**  $(p, q) \in N_\alpha \times N_{\alpha+1}$  **do**
- 12          $\lfloor$  **if** conditions A.1, A.2, A.3, A.4, A.5 are met for  $(p, q)$  **then**
- 13              $\lfloor$   $A = A \cup \{(p, q)\}$
- 14              $\lfloor$  **if**  $W(q) = \mathcal{T}$  **then**
- 15                  $\lfloor$  **return**  $\alpha + 2$
- 16      $\lfloor$  Delete all the vertices  $p \in N_{\alpha+1}$  without predecessor
- 17 **return** False

---

- Since  $\sigma(\mathcal{G})$  is feasible, tasks from  $B(q_{\alpha+1})$  are schedulable at time  $\alpha + 1$  and thus, properties A.2 and A.3 are verified.
- Since  $\sigma(\mathcal{G})$  is an as-soon-as-possible schedule, property A.4 is fulfilled.
- Lastly, since  $\sigma(\mathcal{G})$  satisfies the preferred sons property, A.5 is fulfilled.

We conclude that  $(q_0, q_1, \dots, q_{C-1})$  is a path of  $S(\mathcal{G})$ . Moreover, since  $\sigma(\mathcal{G})$  is of makespan  $C$ ,  $W(q_{C-1}) = \mathcal{T}$ , and thus the lemma is verified.  $\square$

**Lemma 5.** *Let  $C \leq \bar{C}$  and  $(p_0, p_1, \dots, p_{C-1})$  a path of  $S(\mathcal{G})$  with  $W(p_{C-1}) = \mathcal{T}$ . Then, for each task  $i \in \mathcal{T}$ , there exists a unique value  $\alpha \in \{0, \dots, C - 1\}$  such that  $i \in B(p_\alpha)$ .*

*Proof.* According to the definition of  $S(\mathcal{G})$ ,  $W(p_0) \subset W(p_1) \subset \dots \subset W(p_{C-1})$ . Moreover, by assumption,  $W(p_{C-1}) = \mathcal{T}$ . Thus, for each task  $i \in \mathcal{T}$ , there is a unique  $\alpha \in \{0, \dots, C - 1\}$  with  $i \in W(p_\alpha)$  and  $i \notin W(p_{\alpha-1})$ . Since  $W(p_{\alpha-1}) \cup B(p_\alpha) = W(p_\alpha)$ , we get  $i \in B(p_\alpha)$ .  $\square$

**Lemma 6.** *Every path  $(p_0, p_1, \dots, p_{C-1})$  of  $S(\mathcal{G})$  with  $C \leq \bar{C}$  and  $W(p_{C-1}) = \mathcal{T}$  is associated to a feasible schedule of makespan  $C$ .*

*Proof.* Let  $(p_0, p_1, \dots, p_{C-1})$  be a path of  $S(\mathcal{G})$  with  $C \leq \bar{C}$  and  $W(p_{C-1}) = \mathcal{T}$ . A schedule  $\sigma(\mathcal{G})$  of makespan  $C$  is defined as follows:

- By Lemma 5, for any task  $i \in \mathcal{T}$ , there exists a unique value  $\alpha \in \{0, \dots, C - 1\}$  with  $i \in B(p_\alpha)$ . Thus, we set  $t_i^\sigma = \alpha$ .

- For any arc  $(i, j) \in \mathcal{A}$ , we set  $x_{ij}^\sigma = 1$  if  $t_j^\sigma > t_i^\sigma + 1$ , otherwise  $x_{ij}^\sigma = 0$ .

We prove that the schedule  $\sigma(\mathcal{G})$  satisfies the equations of the integer linear program  $P$ , and thus is feasible.

According to the condition A.2, we get  $\Gamma^-(i) \subseteq W(p_{\alpha-1})$ , so  $t_j^\sigma + 1 \leq t_i^\sigma, \forall (j, i) \in \mathcal{A}$ . Following the definition of  $x^\sigma$ , we observe that equations (1) are true.

Now, by definition of  $\sigma(\mathcal{G})$ ,  $t_i^\sigma \leq C - 1, \forall i \in \mathcal{T}$  and thus equations (2) are validated.

According to the condition A.5, for any task  $i \in B(p_\alpha), \alpha \in \{0, \dots, C - 2\}$

C.1 If  $PS_{W(p_\alpha), B(p_\alpha)}(i) \cap X_{\alpha+1} \neq \emptyset$ , then there is exactly one task  $j^* \in \Gamma^+(i) \cap B(p_{\alpha+1})$ , i.e. such that  $t_{j^*}^\sigma = \alpha + 1 = t_i^\sigma + 1$ . The task  $j^*$  is thus the unique successor of  $i$  for which  $x_{ij^*}^\sigma = 0$  and  $\forall j \in \Gamma^+(i) - \{j^*\}, x_{ij}^\sigma = 1$ . Thus,  $\forall i \in \mathcal{T}, \sum_{j \in \Gamma^+(i)} x_{ij}^\sigma = |\Gamma^+(i)| - 1$ .

C.2 If  $PS_{W(p_\alpha), B(p_\alpha)}(i) \cap X_{\alpha+1} = \emptyset$ , then no successor of  $i$  is scheduled at time  $\alpha + 1$ , thus  $\forall j \in \Gamma^+(i), x_{ij}^\sigma = 1$  and  $\sum_{j \in \Gamma^+(i)} x_{ij}^\sigma = |\Gamma^+(i)|$ .

Therefore, equations (3) are checked. Lastly, according to the condition A.3, any task  $i \in B(p_{\alpha+1})$  cannot have more than one predecessor in  $B(p_\alpha)$ , thus  $i$  has at least one predecessor  $j^*$  such that  $x_{j^*i}^\sigma = 0$ . Therefore,  $\forall i \in \mathcal{T}, \sum_{j \in \Gamma^-(i)} x_{ji}^\sigma \geq |\Gamma^-(i)| - 1$  and equations (4) are validated. We conclude that  $\sigma(\mathcal{G})$  is a feasible schedule, and the lemma is proved.  $\square$

**Theorem 2 (Validity of Algorithm 1).** *Algorithm 1 returns the minimum makespan  $C$  of a feasible schedule if  $C \leq \bar{C}$ , false otherwise.*

*Proof.* Let us suppose first that our algorithm returns  $C \leq \bar{C}$ , then the minimum path from  $p_0$  to a node  $p$  with  $W(p) = \mathcal{T}$  is of length  $C$ . By Lemma 6 this path is associated to a feasible schedule of makespan  $C$  and thus this schedule is optimal.

Now, let us suppose that such a path does not exist; in this case, Algorithm 1 returns false. By Lemma 4, there is no feasible schedule of makespan  $C \leq \bar{C}$  and the theorem is proved.  $\square$

## 6 Complexity analysis

We prove in this section that Algorithm 1 is a fixed-parameter algorithm.

**Lemma 7.** *Let us denote by  $n$  the number of tasks and  $pw(\bar{C})$  the pathwidth of the interval graph built with the upper bound  $\bar{C}$  of the minimum makespan. The number of nodes  $|N|$  of the multistage graph  $S(\mathcal{G}) = (N, A)$  is  $\mathcal{O}(n \cdot 2^{2pw(\bar{C})})$  and the number of arcs  $|A|$  is  $\mathcal{O}(n \cdot 2^{4pw(\bar{C})})$ .*

*Proof.* According to Algorithm 1, each node  $p \in N_\alpha$  is such that  $p = (W(p), B(p))$  with  $W(p) = Y(p) \cup Z_\alpha$  and  $Y(p) \subseteq X_\alpha \cap X_{\alpha+1}$ . The number of possibilities for  $Y(p)$  is thus bounded by  $2^{|X_\alpha \cap X_{\alpha+1}|} \leq 2^{|X_\alpha|}$ . Now, since  $B(p) \subseteq X_\alpha$ , the

number of possibilities for  $B(p)$  is bounded by  $2^{|X_\alpha|}$ . Then, the number of nodes in  $N_\alpha$  for  $\alpha \in \{0, \dots, \bar{C} - 1\}$  is bounded by  $2^{2^{|X_\alpha|}}$ .

By definition of the pathwidth, the value  $|X_\alpha|$  is bounded by  $pw(\bar{C}) + 1$ , thus the number of nodes  $|N_\alpha|$  is  $\mathcal{O}(2^{2^{pw(\bar{C})}})$ . Now, since  $\bar{C} \leq n$ , the number of nodes  $|N|$  is  $\mathcal{O}(n \cdot 2^{2^{pw(\bar{C})}})$ . Moreover, the size of  $N_\alpha \times N_{\alpha+1}$  for  $\alpha \in \{0, \dots, \bar{C} - 1\}$  is  $\mathcal{O}(2^{4^{pw(\bar{C})}})$ , thus the whole number of arcs  $|A|$  is  $\mathcal{O}(n \cdot 2^{4^{pw(\bar{C})}})$ , and we get the lemma.  $\square$

**Lemma 8.** *For any  $\alpha \in \{0, \dots, \bar{C} - 2\}$ , the time complexity of checking the conditions A.1 to A.5 for a couple of nodes  $(p, q) \in N_\alpha \times N_{\alpha+1}$  is  $\mathcal{O}(n^2 \cdot pw(\bar{C}))$ .*

*Proof.* The time complexity for checking the condition A.1 is  $\mathcal{O}(n)$ . For the condition A.2, we need to build the set  $\{i \in X_{\alpha+1}, \Gamma^-(i) \subseteq W(p)\}$ . If we denote by  $m$  the number of arcs of  $\mathcal{G}$ , building this set requires to enumerate all the successors of tasks in  $X_{\alpha+1}$ , which is in time complexity equal to  $\mathcal{O}(m)$ . Since  $m \leq n^2$ , the time complexity for checking the condition A.2 is thus  $\mathcal{O}(n^2 \cdot pw(\bar{C}))$ . For the same reasons, time complexity for checking the conditions A.3 and A.4 is also  $\mathcal{O}(n^2 \cdot pw(\bar{C}))$ . For condition the A.5, the time complexity of the computation of the preferred sons of a task  $i \in B(p)$  is also  $\mathcal{O}(n^2)$ , and thus checking this condition also takes  $\mathcal{O}(n^2 \cdot pw(\bar{C}))$ , and the lemma holds.  $\square$

**Theorem 3 (Complexity of Algorithm 1).** *The time complexity of Algorithm 1 is  $\mathcal{O}(n^3 \cdot pw(\bar{C}) \cdot 2^{2^{pw(\bar{C})}})$ , where  $pw(\bar{C})$  is the pathwidth of the interval graph associated to the time windows  $[r_i, d_i]$ ,  $i \in \mathcal{T}$ .*

*Proof.* The time complexity of the computation of the release dates and deadlines (lines 1 – 2) and the sets  $X_\alpha$  and  $Z_\alpha$  for  $\alpha \in \{0, \dots, \bar{C}\}$  (lines 3 – 4) is  $\mathcal{O}(n^2)$  since  $\bar{C}$  is bounded by  $n$ . The time complexity for building  $N$  at lines 7 – 9 is  $\mathcal{O}(n \cdot 2^{2^{pw(\bar{C})}})$  by Lemma 7. Following Lemma 7 and 8, the complexity of building arcs of  $S(\mathcal{G})$  in lines 10 – 17 is  $\mathcal{O}(n^3 \cdot pw(\bar{C}) \cdot 2^{4^{pw(\bar{C})}})$ , thus the theorem holds.  $\square$

## 7 Conclusion

We have shown in this paper that the problem  $\bar{P}|prec, p_i = 1, c_{ij} = 1|C_{max}$  is fixed-parameter tractable. The parameter considered is the pathwidth associated with an upper bound  $\bar{C}$  of the makespan. For this purpose, we have developed a dynamic programming algorithm of complexity  $\mathcal{O}(n^3 \cdot pw(\bar{C}) \cdot 2^{4^{pw(\bar{C})}})$ . This is, as far as we know, the first fixed-parameter algorithm for a scheduling problem with communication delays.

This work opens up several perspectives. The first one is to test experimentally the efficiency of this algorithm, and to compare it to other exact methods such as integer linear programming or dedicated exact methods [20, 17]. A second perspective is to study the extension of this algorithm to more general problems in order to get closer to applications and to evaluate if these approaches can be considered to solve real-life problems.

## References

1. Ait Aba, M., Munier Kordon, A., Pallez, G.: Scheduling on two unbounded resources with communication costs. In: Euro-Par 2019: Parallel Processing - 25th International Conference on Parallel and Distributed Computing, Göttingen, Germany, August 26-30, 2019, Proceedings. pp. 117–128 (2019)
2. Ait El Cadi, A., Ben Atitallah, R., Hanafi, S., Mladenovic, N., Artiba, A.: New MIP model for multiprocessor scheduling problem with communication delays. *Optimization Letters* p. 15 (2014)
3. Andronikos, T., Koziris, N., Papakonstantinou, G., Tsanakas, P.: Optimal scheduling for UET/UET-UCT generalized n-dimensional grid task graphs. *J. Parallel Distrib. Comput.* **57**(2), 140–165 (1999)
4. Bodlaender, H.L.: A tourist guide through treewidth. *Acta Cybern.* **11**, 1–21 (1992)
5. Chrétienne, P.: A polynomial algorithm to optimally schedule tasks on a virtual distributed system under tree-like precedence constraints. *European Journal of Operational Research* **43**(2), 225–230 (1989)
6. Chrétienne, P., Picouleau, C.: Scheduling with communication delays: A survey, pp. 65–90 (1995)
7. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: *Parameterized Algorithms*. Springer Publishing Company, Incorporated, 1st edn. (2015)
8. Davidović, T., Liberti, L., Maculan, N., Mladenovic, N.: Towards the optimal solution of the multiprocessor scheduling problem with communication delays. In: MISTA conference (2007)
9. Downey, R.G., Fellows, M.R.: *Fundamentals of Parameterized Complexity* (2013)
10. Drozdowski, M.: *Scheduling for Parallel Processing*. Springer (2009)
11. Giroudeau, R., Koenig, J.C.: Scheduling with communication delays. In: Levner, E. (ed.) *Multiprocessor Scheduling*, chap. 4. IntechOpen, Rijeka (2007)
12. Graham, R.L., Lawler, E., Lenstra, J., Rinnooy Kan, A.: Optimization and approximation in deterministic sequencing and scheduling: a survey. In: Hammer, P., Johnson, E., Korte, B. (eds.) *Discrete Optimization II*, *Annals of Discrete Mathematics*, vol. 5, pp. 287 – 326. Elsevier (1979)
13. Hoogeveen, J., Lenstra, J., Veltman, B.: Three, four, five, six, or the complexity of scheduling with communication delays. *Operations Research Letters* **16**(3), 129 – 137 (1994)
14. Mnich, M., Van Bevern, R.: Parameterized complexity of machine scheduling: 15 open problems. *Computers & Operations Research* **100** (2018)
15. Munier Kordon, A.: A fixed-parameter algorithm for scheduling unit dependent tasks on parallel machines with time windows. *Discrete Applied Mathematics* **290**, 1–6 (2021)
16. Möhring, R.H., Schäffter, M.W.: Scheduling series–parallel orders subject to 0/1-communication delays. *Parallel Computing* **25**(1), 23 – 40 (1999)
17. Orr, M., Sinnen, O.: Optimal task scheduling benefits from a duplicate-free state-space. *Journal of Parallel and Distributed Computing* **146** (2020)
18. Rayward-Smith, V.: UET scheduling with unit interprocessor communication delays. *Discrete Applied Mathematics* **18**(1), 55–71 (1987)
19. Shimada, K., Taniguchi, I., Tomiyama, H.: Communication-aware scheduling for malleable tasks. In: 2019 International Conference on Platform Technology and Service (PlatCon). pp. 1–6 (2019)

20. Sinnens, O.: Reducing the solution space of optimal task scheduling. *Computers & Operations Research* **43**, 201–214 (2014)
21. Veltman, B., Lageweg, B., Lenstra, J.: Multiprocessor scheduling with communication delays. *Parallel Computing* **16**(2), 173 – 182 (1990)
22. Veltman, B.: Multiprocessor scheduling with communication delays. Ph.D. thesis, Eindhoven University of Technology (1993)
23. Venugopalan, S., Sinnens, O.: ILP formulations for optimal task scheduling with communication delays on parallel systems. *IEEE Transactions on Parallel and Distributed Systems* **26**(1), 142–151 (2015)
24. Zinder, Y., Su, B., Singh, G., Sorli, R.: Scheduling UET-UCT tasks: Branch-and-bound search in the priority space. *Optimization and Engineering* **11**, 627–646 (2010)