



**HAL**  
open science

# A Fixed-Parameter Algorithm for Scheduling Unit Dependent Tasks on Parallel Machines with Time Windows

Alix Munier Kordon

► **To cite this version:**

Alix Munier Kordon. A Fixed-Parameter Algorithm for Scheduling Unit Dependent Tasks on Parallel Machines with Time Windows. *Discrete Applied Mathematics*, 2020. hal-03041735

**HAL Id: hal-03041735**

**<https://hal.science/hal-03041735v1>**

Submitted on 5 Dec 2020

**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 on Parallel Machines with Time Windows

Alix Munier Kordon

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

---

## Abstract

This paper proves that the existence of a feasible schedule for a set of dependent tasks of unit execution times with release dates and deadlines on a limited number of processors is a fixed-parameter tractable problem. The parameter considered is the pathwidth of the interval graph associated with the time windows of the tasks. A fixed-parameter algorithm based on a dynamic programming approach is developed and proved to solve this decision problem.

Fixed-parameter algorithms for the two classical problems  $P|prec, p_i = 1, r_i|C_{\max}$  and  $P|prec, p_i = 1, r_i|L_{\max}$  are then derived using a binary search. They are, as far as we know, the first fixed-parameter algorithms for these scheduling problems.

*Keywords:* parallel identical machines, makespan, maximum lateness, parameterized complexity

---

## 1. Introduction

This paper tackles a basic scheduling problem defined as follows: we consider a set  $T$  of  $n$  tasks of unit duration. Precedence relations are expressed by an (acyclic) precedence graph noted  $G = (T, A)$ . If  $s_i^\sigma$  denotes the starting time of a task  $i \in T$  according to a feasible schedule  $\sigma$ , each arc  $e = (i, j) \in A$  expresses the precedence constraint  $s_i^\sigma + 1 \leq s_j^\sigma$ . Moreover, each task must be executed during its time window, i.e. for each task  $i \in T$ ,  $r_i \leq s_i^\sigma < d_i$  where  $r_i$  is the release time of  $i$  and  $d_i$  its deadline. We assume without loss of generality that  $\min_{i \in T} r_i = 0$ . Lastly,  $m$  identical machines are available,

10 so that at most  $m$  tasks can be scheduled at each time instant. The ques-  
 11 tion is the existence of a feasible schedule. This problem is referred to as  
 12  $P|prec, p_i = 1, r_i, d_i|*$  using standard notations [11].

13 Let  $C(\sigma) = \max_{i \in T}(s_i^\sigma + 1)$  be the makespan of a schedule  $\sigma$  and  $L(\sigma) =$   
 14  $\max_{i \in T}(\max(0, s_i^\sigma + 1 - d_i))$  its maximum lateness. Let us consider the op-  
 15 timization problem  $P|prec, p_i = 1, r_i|C_{\max}$ . For each upper bound  $C \geq 0$   
 16 of the makespan, the deadline of each task  $i \in T$  can be set to  $d_i = C - \ell_i$   
 17 where  $\ell_i$  is the maximum number of arcs of a path starting from  $i$ . We ob-  
 18 serve that any solution of the corresponding instance of  $P|prec, p_i = 1, r_i, d_i|*$   
 19 is a feasible solution of  $P|prec, p_i = 1, r_i|C_{\max}$ . We conclude that any algo-  
 20 rithm solving the decision problem  $P|prec, p_i = 1, r_i, d_i|*$  can be combined  
 21 with binary search to solve  $P|prec, p_i = 1, r_i|C_{\max}$ . A similar remark can  
 22 be made for the optimisation problem  $P|prec, p_i = 1, r_i|L_{\max}$ . Indeed, for  
 23 each instance of this problem and each upper bound  $\lambda \geq 0$  of the maximum  
 24 lateness, we can set the deadline of each task  $i \in T$  to  $d'_i = d_i + \lambda$  and solve  
 25 the associated decision problem  $P|prec, p_i = 1, r_i, d'_i|*$ . So, each algorithm  
 26 solving the decision problem  $P|prec, p_i = 1, r_i, d_i|*$  can thus be extended for  
 27  $P|prec, p_i = 1, r_i|C_{\max}$  or  $P|prec, p_i = 1, r_i|L_{\max}$  by adding a binary search  
 28 on the upper bound of respectively the makespan or the maximum lateness.

29 The optimisation problem  $P|prec, p_i = 1|C_{\max}$  has been widely studied. It  
 30 was proved to be NP-hard initially for opposing forests by Garey et al. [10].  
 31 The recent survey of Prot and Bellenguez-Morineau [15] presents several  
 32 complexity results depending on the structure of the precedence graph and  
 33 the number of processors.

34 The development of fixed-parameter algorithms for NP-complete prob-  
 35 lems is a way to get polynomial-time algorithms when some parameters are  
 36 fixed [6, 8]. More formally, a fixed-parameter algorithm solves any instance  
 37 of the problem of size  $n$  in time  $f(k) \times \text{poly}(n)$ , where  $f$  is allowed to be a  
 38 computable superpolynomial function and  $k$  the associated parameter.

39 Several authors developed exact algorithms for  $P|prec, p_i = 1|C_{\max}$ . Dolev  
 40 and Warmuth [7] developed a dynamic programming algorithm of time com-  
 41 plexity  $\mathcal{O}(n^{h(G)(m-1)+1})$  to solve this problem. The parameter  $h(G)$  repre-  
 42 sents the number of nodes of the longest path of the precedence graph  $G$ .  
 43 Few years later, Möhring [14] developed another dynamic programming al-  
 44 gorithm of time complexity  $\mathcal{O}(n^{w(G)})$ . The parameter  $w(G)$  is the width of  
 45 the precedence graph  $G$  defined as the size of its largest antichain. None of  
 46 these algorithms are fixed-parameter ones.

47 Mnich and van Bevern [13] surveyed main results on parameterized com-

48 plexity for scheduling problems and identified 15 open problems. They point  
49 out that the fixed-parameter tractability of  $P3|prec, p_i = 1|C_{\max}$  param-  
50 eterized by the width  $w(G)$  of the precedence graph  $G$  is an open chal-  
51 lenging question. The width  $w(G)$  leads to several negative results. Du et  
52 al. [9] proved that  $P2|chains|C_{\max}$  is strongly NP-hard for unbounded width.  
53 Günther et al. [12] proved that  $P2|chains, w(G) \leq 3|C_{\max}$  is weakly NP-hard.  
54 Bodlaender and Fellows [4] proved that  $P|prec, p_i = 1|C_{\max}$  is W[2]-hard pa-  
55 rameterized by the width and the number of machines. More recently, van  
56 Bevern et al. [2] proved that  $P2|prec, p_j \in \{1, 2\}|C_{\max}$  is W[2]-hard parame-  
57 terized by the width  $w(G)$ , and that this result is an argument to look at the  
58 parameterized complexity of  $P3|prec, p_i = 1|C_{\max}$  using  $w(G)$  as parameter.  
59 The existence of a fixed-parameter algorithm for the optimisation problem  
60  $P|prec, p_i = 1|C_{\max}$  is a challenging question, but as we described above,  
61 most of the authors focused their study on the width  $w(G)$  as a parameter.

62 The parameter considered in this paper is the pathwidth  $pw(\mathcal{I})$  of an  
63 instance  $\mathcal{I}$  of the decision problem  $P|prec, p_i = 1, r_i, d_i|\star$ . Let us consider the  
64 strictly increasing sequences  $u_\alpha$ ,  $\alpha \in \mathbb{N}$  corresponding to all the time instants  
65 of the beginnings and endings of the intervals  $[r_i, d_i]$ ,  $i \in T$ . By definition of  
66 release dates,  $u_0 = 0$ . Moreover, the last element  $u_{k^*}$  of the sequence satisfies  
67  $k^* \leq 2n - 1$ . Then, for any  $\alpha \in \{1, \dots, k^*\}$ , we set  $X_\alpha = \{i \in T, (u_{\alpha-1}, u_\alpha) \cap$   
68  $(r_i, d_i) \neq \emptyset\}$ . The pathwidth is defined as  $pw(\mathcal{I}) = \max_{\alpha \in \{1, \dots, k^*\}} (|X_\alpha| - 1)$ .

69 The pathwidth  $pw(\mathcal{I})$  is a simple measure of the parallelism of the in-  
70 stance  $\mathcal{I}$ . One can observe that  $pw(\mathcal{I})$  is the pathwidth of the interval graph  
71 associated with the set of intervals  $\{(r_i, d_i), i \in T\}$  [3]. As far as we know,  
72 this measure was not yet considered as a parameter for a scheduling problem.

73 We prove in this paper that the decision problem  $P|prec, p_i = 1, r_i, d_i|\star$   
74 can be solved in time  $\mathcal{O}(n^2 2^{4 \times pw(\mathcal{I})})$  using a dynamic programming approach  
75 based on a multistage auxiliary graph which longest paths model all the  
76 feasible schedules. By coupling our algorithm with a binary search based  
77 on a lower bound of the respective objectives, we get that the two opti-  
78 misation problems  $P|prec, p_i = 1, r_i|C_{\max}$  and  $P|prec, p_i = 1, r_i|L_{\max}$  are  
79 fixed-parameter tractable parameterized by the pathwidth.

80 Baptiste and Timkovsky [1] developed a similar approach to get a polyno-  
81 mial time algorithm of time complexity  $\mathcal{O}(n^9)$  for  $P2|prec, p_i = 1, r_i|\sum C_j$ .  
82 They also conjectured the existence of ideal schedules, i.e. that minimize both  
83 the makespan and the total completion time for the previous instance. More  
84 recently, Coffman et al. [5] proved that each optimal solution of  $P2|prec, p_i =$   
85  $1, r_i|\sum C_j$  minimizes the makespan of the same instance and they improved

86 the algorithm of Baptiste and Timkowsky to get a complexity in time  $\mathcal{O}(n^3)$ .

87 This paper is organised as follows: Section 2 presents additional notations  
 88 and an example. Section 3 is devoted to the description of properties of  
 89 feasible schedules. These properties will be considered in Section 4 to build  
 90 the multistage auxiliary graph  $\mathcal{G}$  whose longest paths represent all the feasible  
 91 solutions (if any). Section 5 details the complexity of the construction of  $\mathcal{G}$ .  
 92 Section 6 is our conclusion.

## 93 2. Additional notations and example

94 For any task  $i \in T$ , the set of immediate successors and predecessors of  $i$  is  
 95 respectively denoted by  $\Gamma^+(i)$  and  $\Gamma^-(i)$ . For any couple of tasks  $(i, j) \in T^2$ ,  
 96 we denote by  $i \rightarrow j$  if there is a path in  $G$  from  $i$  to  $j$ . The descendants of  
 97 a task  $i \in T$ , denoted by  $\Gamma^{+\star}(i)$  is the set of tasks  $j \neq i$  such that  $i \rightarrow j$ .  
 98 Similarly,  $\Gamma^{-\star}(i)$  denotes the set of ancestors of  $i$ , i.e. the set of tasks  $j \neq i$   
 99 such that  $j \rightarrow i$ .

100 We assume without loss of generality that release dates and deadlines  
 101 are consistent with respect to precedence constraints, i.e. if  $(i, j) \in A$   
 102 then  $r_i + 1 \leq r_j$  and  $d_i + 1 \leq d_j$ . For any task  $i \in T$ ,  $f(i)$  is the small-  
 103 est value  $\alpha \in \{1, \dots, k^\star\}$  such that  $i \in X_\alpha$ .

104 Figure 1 presents an instance  $\mathcal{I}$  of our scheduling problem for two ma-  
 105 chines. We get  $k^\star = 5$  and  $u_\alpha = \alpha$  for  $\alpha \in \{0, \dots, 5\}$ . The corresponding  
 106 sets are  $X_1 = \{1\}$ ,  $X_2 = \{1, 2, 4\}$ ,  $X_3 = \{2, 3, 4, 5, 6\}$ ,  $X_4 = \{3, 5, 6, 7\}$  and  
 107  $X_5 = \{7\}$ . The pathwidth is  $pw(\mathcal{I}) = |X_3| - 1 = 4$ . We also have  $f(1) = 1$ ,  
 $f(2) = f(4) = 2$ ,  $f(3) = f(5) = f(6) = 3$  and  $f(7) = 4$ .

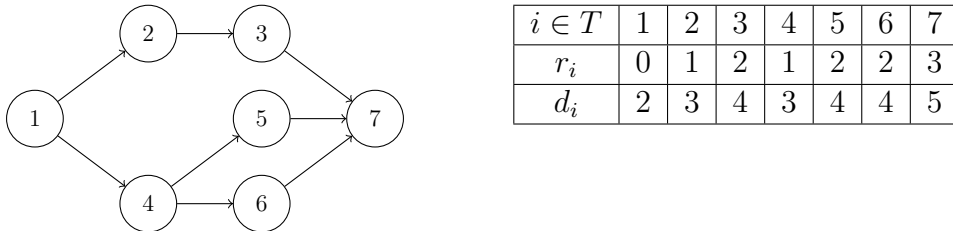


Figure 1: An instance  $\mathcal{I}$  of  $P|prec, p_i = 1, r_i, d_i| \star$  for  $m = 2$  machines.

108

109 **3. Basic properties of feasible solutions**

110 Let us suppose that  $\sigma$  is a feasible solution of our scheduling problem. For  
 111 any value  $\alpha \in \{1, \dots, k^*\}$ , we define the slice  $S_\alpha = \{i \in T, u_{\alpha-1} \leq s_i^\sigma < u_\alpha\}$ .

112 Consider the feasible schedule  $\sigma$  for our example given in Figure 2. We  
 113 get the slices  $S_1 = \{1\}$ ,  $S_2 = \{2, 4\}$ ,  $S_3 = \{3, 5\}$ ,  $S_4 = \{6\}$  and  $S_5 = \{7\}$ .



Figure 2: A feasible schedule  $\sigma$  associated with the example given in Figure 1.

114 **Definition 1.** A set of tasks  $V \subseteq T$  is said to be consistent if for any couple  
 115  $(i, j) \in V^2$  with  $i \neq j$  such that  $i \rightarrow j$ , any task  $k$  of a path in  $G$  from  $i$  to  $j$   
 116 belongs to  $V$ .

117 **Definition 2.** A set  $V \subseteq T$  is said to be strictly consistent if for any  $x \in V$ ,  
 118  $\Gamma^+(x) \subseteq V$ .

119 We observe that any strictly consistent set  $V$  is consistent, but the reverse  
 120 is not true. The consequence is that Definition 2 is more restrictive than  
 121 Definition 1.

122 **Lemma 1.** For any value  $\alpha \in \{1, \dots, k^*\}$ ,  $S_\alpha$  is consistent.

123 *Proof.* Suppose, for the sake of contradiction, that  $S_\alpha$  is not consistent. Then,  
 124 there exist a couple  $(i, j) \in S_\alpha^2$  and a task  $k \in S_\beta$  with  $\beta \neq \alpha$  and  $i \rightarrow k \rightarrow j$ .  
 125 If  $\beta > \alpha$ , then the precedence  $k \rightarrow j$  cannot be fulfilled since  $s_k^\sigma \geq u_\alpha > s_j^\sigma$ .  
 126 Similarly, if  $\beta < \alpha$ , the precedence  $i \rightarrow k$  cannot be fulfilled as  $s_k^\sigma < u_{\alpha-1} \leq$   
 127  $s_i^\sigma$ , thus the lemma is proved.  $\square$

128 **Lemma 2.** For any value  $\alpha \in \{1, \dots, k^*\}$ ,  $\cup_{\beta=\alpha}^{k^*} S_\beta$  is strictly consistent.

129 *Proof.* Suppose, for the sake of contradiction, that there exists  $\alpha \in \{1, \dots, k^*\}$   
 130 such that  $\cup_{\beta=\alpha}^{k^*} S_\beta$  is not strictly consistent. Let then  $i \in \cup_{\beta=\alpha}^{k^*} S_\beta$  with  
 131  $\Gamma^+(i) \not\subseteq \cup_{\beta=\alpha}^{k^*} S_\beta$  and  $j \in \Gamma^+(i) - \cup_{\beta=\alpha}^{k^*} S_\beta$ . Thus,  $j \in S_\beta$  with  $\beta < \alpha$ ,  
 132 and then  $s_j^\sigma < u_{\alpha-1} \leq s_i^\sigma$ . The precedence  $i \rightarrow j$  is not fulfilled by  $\sigma$ ,  
 133 a contradiction.  $\square$

134 **Lemma 3.** For any  $\alpha \in \{1, \dots, k^*\}$ , let  $Z_\alpha = \{i \in T, f(i) \geq \alpha\}$ . If  $\sigma$   
135 is a feasible schedule, then any subset  $V_\alpha = \cup_{\beta=\alpha}^{k^*} S_\beta$  can be partitioned such as  
136  $V_\alpha = Z_\alpha \cup p(V)$  with  $p(V) \subseteq X_\alpha$ .

137 *Proof.* Any element  $i \in Z_\alpha$  satisfies  $f(i) \geq \alpha$  and thus  $r_i \geq u_{\alpha-1}$ . Since  $\sigma$   
138 is feasible,  $s_i^\sigma \geq r_i$ , and thus  $i \in V_\alpha$ . We conclude that  $Z_\alpha \subseteq V_\alpha$ .

139 Now, let  $i \in p(V_\alpha) = V_\alpha - Z_\alpha$ , then  $f(i) < \alpha$ , thus  $r_i < u_{\alpha-1}$ . Then,  
140 there exists  $\beta < \alpha$  with  $i \in X_\beta$ . Since  $i \in V_\alpha$ , there exists  $\beta' \in \{\alpha, \dots, k^*\}$   
141 with  $i \in S_{\beta'}$ , and then  $s_i^\sigma \leq u_{\beta'} \leq d_i$ . Since  $\beta' \geq \alpha$ ,  $u_{\beta'} \geq u_\alpha$ . As  $r_i < u_{\alpha-1}$   
142 and  $d_i \geq u_\alpha$ ,  $(r_i, d_i) \cap (u_{\alpha-1}, u_\alpha) \neq \emptyset$  and thus  $i \in X_\alpha$ , which concludes  
143 the proof.  $\square$

144 For the example given by Figure 1, we get  $Z_5 = \emptyset$ ,  $Z_4 = \{7\}$ ,  $Z_3 =$   
145  $\{3, 5, 6, 7\}$ ,  $Z_2 = \{2, 3, 4, 5, 6, 7\}$  and  $Z_1 = T$ . For the schedule  $\sigma$  from  
146 Figure 2,  $V_1 = T = Z_1$ ,  $V_2 = \{2, 3, 4, 5, 6, 7\} = Z_2$ ,  $V_3 = \{3, 5, 6, 7\} = Z_3$   
147  $V_4 = \{6, 7\} = \{6\} \cup Z_4$  and  $V_5 = \{7\} = \{7\} \cup Z_5$ . We deduce that  
148  $p(V_1) = p(V_2) = p(V_3) = \emptyset$ ,  $p(V_4) = \{6\}$  and  $p(V_5) = \{7\}$ .

#### 149 4. Representation of feasible solutions and description of the algo- 150 rithm

151 Let  $\mathcal{I}$  be an instance of  $P|prec, p_i = 1, r_i, d_i|*$ . The idea of the algorithm  
152 is to build a multistage auxiliary graph  $\mathcal{G} = (\mathcal{T}, \mathcal{A})$  whose longest paths  
153 describe all the feasible solutions (if any) of  $\mathcal{I}$ .

154 Elements of  $\mathcal{T}$  are associated with subsets of  $T$ . The set  $\mathcal{T}$  is partitioned  
155 into  $k^*$  sets  $\mathcal{T}_1, \dots, \mathcal{T}_{k^*}$  such that, for any value  $\alpha \in \{1, \dots, k^*\}$ , elements  
156 of  $\mathcal{T}_\alpha$  are strictly consistent subsets of  $X_\alpha - Z_\alpha$ . Then, every element  $v \in \mathcal{T}_\alpha$   
157 verifies  $v \cup Z_\alpha \subseteq X_\alpha \cup Z_\alpha \subseteq \bigcup_{\beta=\alpha}^{k^*} X_\beta$ . Moreover,  $p(v \cup Z_\alpha) = v$ .

158 Now, let us suppose that  $v$  and  $v'$  are two vertices respectively of  $\mathcal{T}_\alpha$  and  
159  $\mathcal{T}_{\alpha+1}$  for  $\alpha \in \{1, \dots, k^* - 1\}$ . Then, there is an arc  $(v, v')$  in  $\mathcal{A}$  if:

- 160 1.  $v' \cup Z_{\alpha+1} \subseteq v \cup Z_\alpha$  and  $S = v \cup Z_\alpha - (v' \cup Z_{\alpha+1})$  is in  $X_\alpha$  and is consistent;
- 161 2. There exists a schedule of tasks from  $S$  on  $m$  machines of makespan  
162  $C_\alpha = u_\alpha - u_{\alpha-1}$  meeting the precedence constraints induced by the sub-  
163 graph  $G' = (S, A)$  of  $G$ .

164 Figure 3 presents the multistage auxiliary graph  $\mathcal{G} = (\mathcal{T}, \mathcal{A})$  associated with  
165 the instance given in Figure 1.

166 We prove in Lemma 4 that any feasible schedule  $\sigma$  is associated with  
167 a path of the multistage auxiliary graph  $\mathcal{G}$  from a vertex of  $\mathcal{T}_1$  to a vertex

168 of  $\mathcal{T}_{k^*}$ . Indeed, let  $S_1, \dots, S_{k^*}$  be the slices of  $\sigma$  and the corresponding se-  
 169 quence  $V_\alpha = \cup_{\beta=\alpha}^{k^*} S_\beta$ , for each  $\alpha \in \{1, \dots, k^*\}$ . Then,  $\sigma$  is associated with  
 170 the path  $p(V_1), p(V_2), \dots, p(V_{k^*})$  of  $\mathcal{G}$ .

171 Let us consider for example the schedule  $\sigma$  given in Figure 2. Recall that  
 172  $p(V_1) = p(V_2) = p(V_3) = \emptyset$ ,  $p(V_4) = \{6\}$  and  $p(V_5) = \{7\}$ . The schedule  $\sigma$   
 173 is thus associated with the path  $\emptyset \rightarrow \emptyset \rightarrow \emptyset \rightarrow \{6\} \rightarrow \{7\}$ .

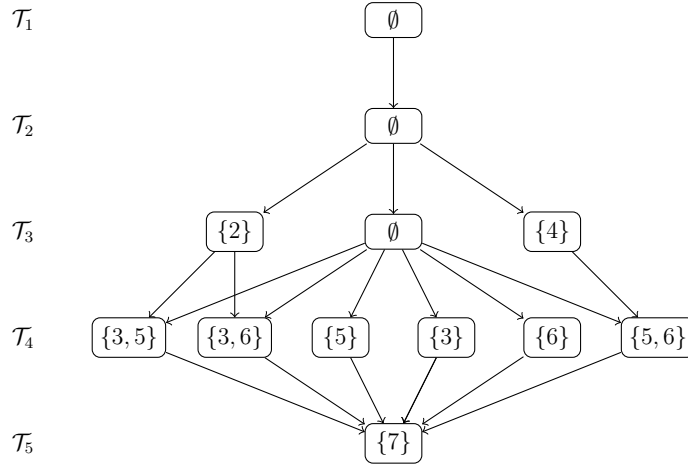


Figure 3: The multistage auxiliary graph  $\mathcal{G} = (\mathcal{T}, \mathcal{A})$  associated with the example given in Figure 1. Vertices without successor are removed.

174 **Lemma 4.** *Each feasible solution  $\sigma$  of the instance  $\mathcal{I}$  is associated with a path*  
 175 *of the multistage auxiliary graph  $\mathcal{G}$  from a vertex of  $\mathcal{T}_1$  to a vertex of  $\mathcal{T}_{k^*}$ .*

176 *Proof.* Let  $S_1, \dots, S_{k^*}$  be the slices of  $\sigma$  and  $V_\alpha = \cup_{\beta=\alpha}^{k^*} S_\beta$  for each  $\alpha \in$   
 177  $\{1, \dots, k^*\}$ . For any value  $\alpha \in \{1, \dots, k^*\}$ ,  $V_\alpha \subseteq \cup_{\beta=\alpha}^{k^*} X_\beta$ .  $V_\alpha$  is strictly  
 178 consistent by Lemma 2 and  $Z_\alpha \subseteq V_\alpha$  by Lemma 3. Thus,  $p(V_\alpha) \in \mathcal{T}_\alpha$ .

179 Now, for any value  $\alpha \in \{1, \dots, k^* - 1\}$ ,  $V_{\alpha+1} \subseteq V_\alpha$ , and the set  $S_\alpha =$   
 180  $V_\alpha - V_{\alpha+1} \subseteq X_\alpha$  is consistent by Lemma 1. Lastly, as  $\sigma$  is a feasible solution,  
 181  $S_\alpha$  can be scheduled according to the resource constraints and the precedence  
 182 expressed by the sub-graph  $(S_\alpha, \mathcal{A})$ . Thus the arc  $(p(V_\alpha), p(V_{\alpha+1})) \in \mathcal{A}$ ,  
 183 which concludes the proof.  $\square$

184 Let us consider now a path  $p(V_1), \dots, p(V_{k^*})$  of  $\mathcal{G}$ . We prove in Lemma 5  
 185 that the sequence of slices  $S_{k^*} = p(V_{k^*}) \cup Z_{k^*}$  and for each  $\alpha \in \{1, \dots, k^* - 1\}$ ,  
 186  $S_\alpha = p(V_\alpha) \cup Z_\alpha - (p(V_{\alpha+1}) \cup Z_{\alpha+1})$  defines a feasible schedule.



187 Let us consider for example the path of the graph  $\mathcal{G}$  given by Figure 3 from  
188  $p(V_1)$  from  $p(V_5)$  such that  $p(V_1) = p(V_2) = \emptyset$ ,  $p(V_3) = \{2\}$ ,  $p(V_4) = \{3, 6\}$   
189 and  $p(V_5) = \{7\}$ . Corresponding slices are  $S_5 = \{7\}$ ,  $S_4 = \{3, 6, 7\} -$   
190  $\{7\} = \{3, 6\}$ ,  $S_3 = \{2, 3, 5, 6, 7\} - \{3, 6, 7\} = \{2, 5\}$ ,  $S_2 = \{2, 3, 4, 5, 6, 7\} -$   
191  $\{2, 3, 5, 6, 7\} = \{4\}$ , and  $S_1 = T - \{2, 3, 4, 5, 6, 7\} = \{1\}$ . We get the sched-  
192 ule  $\sigma$  defined as  $s_1^\sigma = 0$ ,  $s_4^\sigma = 1$ ,  $s_2^\sigma = s_5^\sigma = 2$ ,  $s_3^\sigma = s_6^\sigma = 3$  and  $s_7^\sigma = 4$ .

193 **Lemma 5.** *Each path of the multistage auxiliary graph  $\mathcal{G}$  from a vertex of  $\mathcal{T}_1$*   
194 *to a vertex of  $\mathcal{T}_{k^*}$  is associated with a feasible solution.*

195 *Proof.* Let  $p(V_1), \dots, p(V_{k^*})$  be a path of  $\mathcal{G}$  with  $p(V_\alpha) \in \mathcal{T}_\alpha$ . Let us de-  
196 fine the sequences  $V_\alpha = p(V_\alpha) \cup Z_\alpha$  for  $\alpha \in \{1, \dots, k^*\}$ ,  $S_{k^*} = V_{k^*}$  and  
197 for each  $\alpha \in \{1, \dots, k^* - 1\}$ ,  $S_\alpha = V_\alpha - V_{\alpha+1}$ .

198 By definition of  $\mathcal{G}$ , each set  $S_\alpha$  is consistent and can be scheduled fol-  
199 lowing resource and precedence constraints of the sub-graph  $(S_\alpha, A)$  within  
200 a makespan  $C_\alpha = u_\alpha - u_{\alpha-1}$ . We prove that the schedule obtained by merging  
201 the schedules of the successive slices  $S_1, \dots, S_{k^*}$  is feasible.

202 By definition,  $Z_1 \subseteq V_1$  and  $Z_1 = \{i \in T, f(i) \geq 1\} = T$ , thus  $V_1 =$   
203  $T$ . Moreover, for each arc  $(p(V_\alpha), p(V_{\alpha+1}))$  of  $\mathcal{G}$  with  $\alpha \in \{1, \dots, k^* - 1\}$ ,  
204  $p(V_{\alpha+1}) \cup Z_{\alpha+1} \subseteq p(V_\alpha) \cup Z_\alpha$ , thus  $V_{\alpha+1} \subseteq V_\alpha$  and then  $V_{k^*} \subseteq V_{k^*-1} \subseteq \dots \subseteq$   
205  $V_1$ . Then, each task  $i \in T$  belongs to exactly one slice  $S_\alpha$  and  $S_1, \dots, S_{k^*}$   
206 is a partition of  $T$ .

207 By definition,  $S_\alpha \subseteq X_\alpha$ , thus for any task of  $S_\alpha$ , release dates and  
208 deadlines are fulfilled. Resource constraints are fulfilled by definition of  
209 each set  $S_\alpha$ , for each  $\alpha \in \{1, \dots, k^*\}$ .

210 Lastly let us consider two tasks  $(i, j) \in T^2$  with  $(i, j) \in A$ . Let suppose  
211 that  $i \in S_\alpha$  and  $j \in S_\beta$ . If  $\alpha = \beta$ , the constraint is fulfilled since there  
212 exists a feasible schedule of  $S_\alpha$  for the sub-graph  $(S_\alpha, A)$ . If  $\alpha < \beta$ , the  
213 set  $S_\alpha$  is scheduled before  $S_\beta$ , the constraint is then fulfilled. Lastly, if  
214  $\alpha > \beta$ , then  $V_\alpha$  is not strictly consistent since  $i \in V_\alpha$  and  $j \notin V_\alpha$ , which  
215 is a contradiction. Thus, the lemma is proved.  $\square$

216 The next theorem is a simple outcome of Lemmas 4 and 5.

217 **Theorem 1.** *There exists a feasible schedule of makespan  $C$  if and only if*  
218 *there exists a path in the multistage auxiliary graph  $\mathcal{G}$  from a vertex of  $\mathcal{T}_1$  to*  
219 *a vertex of  $\mathcal{T}_{k^*}$ .*

220 Our algorithm simply builds the multistage auxiliary graph  $\mathcal{G}$  starting  
221 from  $\mathcal{T}_{k^*}$  to  $\mathcal{T}_1$ . Vertices without successor from  $\mathcal{T}_\alpha$  with  $\alpha \neq k^*$  are removed.  
222 In the next section, we investigate the time complexity for computing  $\mathcal{G}$ .

223 **5. Complexity for the computation of the multistage auxiliary graph**

224 Lemmas 6 and 7 study the complexity of two specific tests considered by  
 225 our previous algorithm.

226 **Lemma 6.** *Checking the consistency of a set  $V \subseteq T$  can be done in time  $\mathcal{O}(n^3)$   
 227 and checking the strict consistency of a set  $V \subseteq T$  can be done in time  $\mathcal{O}(n^2)$   
 228 (with  $n = |T|$ ).*

229 *Proof.* For any task  $x \in T$ , the computation of the sets of descendants  $\Gamma(x)^{+\star}$   
 230 and of ancestors  $\Gamma(x)^{-\star}$  can be done once in time  $\mathcal{O}(n^2)$  using a depth-first  
 231 search algorithm. Checking that  $V$  is consistent can be done by checking  
 232 that, for any couple  $(i, j) \in V^2$  with  $i \neq j$ ,  $\Gamma(i)^{+\star} \cap \Gamma(j)^{-\star} \subseteq V$ . The overall  
 233 complexity is then  $\mathcal{O}(n^3)$ .

Similarly, checking that for any task  $x \in V$ ,  $\Gamma^+(x) \subseteq V$  can be done in  
 time  $\mathcal{O}(|\Gamma^+(x)|)$ . The overall complexity is bounded by

$$\sum_{x \in V} |\Gamma^+(x)| \leq \sum_{x \in T} |\Gamma^+(x)| \leq n^2,$$

234 the complexity of the algorithm is then  $\mathcal{O}(n^2)$ . □

235 **Lemma 7.** *The complexity of checking the existence of a feasible schedule  
 236 of fixed duration  $C$  on  $m$  machines for a consistent set of tasks  $V \subseteq T$  with  
 237 the precedence relations modelled by  $G' = (V, A)$  is  $\mathcal{O}(|V|^3 \times 2^{2|V|})$ .*

238 *Proof.* Let us denote by  $\mathcal{P}(V)$  the set of consistent subsets of  $V$ . Let us  
 239 consider the directed graph  $H = (\mathcal{P}(V), E)$  which arcs are built as follows:  
 240 an arc  $(v, v') \in E$  if  $v \subsetneq v'$ ,  $|v' - v| \leq m$  and for any task  $i \in v' - v$ ,  $\Gamma_{G'}^{\star}(i) \subseteq v$   
 241 where  $\Gamma_{G'}^{\star}(i)$  is the set of the ancestors of  $i$  in  $G'$ .

242 Paths of  $H$  from  $\emptyset$  to  $V$  are associated to feasible schedules. Indeed, let  
 243 us consider a path  $v_0 = \emptyset, v_1, \dots, v_p = V$  of  $H$ . For any value  $\beta \in \{1, \dots, p\}$ ,  
 244  $Y_\beta = v_\beta - v_{\beta-1}$  are executed at time  $\beta - 1$ . Conversely any feasible schedule  
 245 corresponds to a path of  $H$ . Thus, there exists a feasible schedule of makespan  
 246 at most  $C$  if the value of a shortest path (considering each arc valued by 1)  
 247 from  $\emptyset$  to  $V$  is inferior than or equal to  $C$ .

248 The number of vertices of  $H$  satisfies  $|\mathcal{P}(V)| \leq 2^{|V|}$ . By Lemma 6, check-  
 249 ing that any element  $v \in \mathcal{P}(V)$  is consistent can be done in time  $\mathcal{O}(|V|^3 \times$   
 250  $2^{|V|})$ . Checking that  $e = (v, v') \in E$  has a time complexity bounded by  
 251  $\mathcal{O}(|V|^2)$  thus the construction of  $H$  can be done in time  $\mathcal{O}(|V|^3 \times 2^{2|V|})$ .

252 Now, any arc  $e = (v, v') \in E$  verifies  $v \subsetneq v'$  and thus  $|v| < |v'|$ . Then,  $H$   
 253 has no circuit. The shortest path can be computed from a topological sort  
 254 in time complexity bounded by  $\mathcal{O}(2^{2|V|})$ , thus the lemma holds.  $\square$

255 **Lemma 8.** *For any instance  $\mathcal{I}$ , the number of vertices  $|\mathcal{T}|$  of the multi-*  
 256 *stage auxiliary graph  $\mathcal{G}$  is in  $\mathcal{O}(n2^{pw(\mathcal{I})})$ , while the number of arcs  $|\mathcal{A}|$  is in*  
 257  *$\mathcal{O}(n2^{2pw(\mathcal{I})})$ .*

258 *Proof.* By Lemma 3, for any value  $\alpha \in \{1, \dots, k^*\}$ , each element  $v \in \mathcal{T}_\alpha$   
 259 is included in  $X_\alpha$  thus  $|\mathcal{T}_\alpha| \leq 2^{|X_\alpha|} \leq 2^{pw(\mathcal{I})+1}$ . Then,  $|\mathcal{T}| = \sum_{\alpha=1}^{k^*} |\mathcal{T}_\alpha| \leq$   
 260  $k^* \times 2^{pw(\mathcal{I})+1}$ . Since  $k^* \leq 2n$ , we get  $|\mathcal{T}| \leq 2n2^{pw(\mathcal{I})+1}$ .

261 Now, the number of arcs of  $\mathcal{G}$  verifies  $|\mathcal{A}| \leq \sum_{\alpha=1}^{k^*-1} (|\mathcal{T}_\alpha| \times |\mathcal{T}_{\alpha+1}|) \leq$   
 262  $k^*(2^{pw(\mathcal{I})+1})^2 \leq 2n \times 2^{2pw(\mathcal{I})+2}$  which concludes the lemma.  $\square$

263 We are now ready to prove our main theorem:

264 **Theorem 2.** *Computing the graph  $\mathcal{G}$  is in time  $\mathcal{O}(n^4 2^{4pw(\mathcal{I})})$  for any in-*  
 265 *stance  $\mathcal{I}$ .*

266 *Proof.* Let us consider all the steps for the construction of the multistage  
 267 auxiliary graph  $\mathcal{G}$  for an instance  $\mathcal{I}$  of the scheduling problem with  $n = |\mathcal{T}|$ .  
 268 The computation of sets  $X_\alpha$  and  $Z_\alpha$  for  $\alpha \in \{1, \dots, k^*\}$  can be done in  
 269 time complexity  $\mathcal{O}(n \log n)$ . For the definition of sets  $\mathcal{T}_\alpha$ ,  $\alpha = 1, \dots, k^*$ , we  
 270 observe that for each subset  $v \subseteq X_\alpha$ , we have to test if  $v \cup Z_\alpha$  is strictly  
 271 consistent, which requires  $\mathcal{O}(n^2)$  instructions by Lemma 6. By Lemma 8,  
 272 the construction of  $\mathcal{T}$  is thus in time  $\mathcal{O}(n^3 \times 2^{pw(\mathcal{I})})$ .

273 Now, for any couple  $(v, v') \in \mathcal{T}_\alpha \times \mathcal{T}_{\alpha+1}$ , the consistency of  $S = (v \cup$   
 274  $Z_\alpha) - (v' \cup Z_{\alpha+1})$  is checked in time  $\mathcal{O}(n^3)$  by Lemma 6. The existence  
 275 of a schedule of  $S$  on  $m$  machines within a makespan  $C_\alpha = u_\alpha - u_{\alpha-1}$  is  
 276 in time  $\mathcal{O}(|S|^3 \times 2^{2|S|})$  following Lemma 7. Now, since  $S \subseteq X_\alpha$ ,  $|S| \leq$   
 277  $pw(\mathcal{I}) + 1$  and thus checking the existence of an arc  $(v, v')$  in  $\mathcal{A}$  is in time  
 278  $\mathcal{O}(n^3 + pw(\mathcal{I})^3 \times 2^{2(pw(\mathcal{I}))})$ . Following Lemma 8, the overall complexity for  
 279 the determination of  $\mathcal{A}$  is then  $\mathcal{O}(n \times 2^{2pw(\mathcal{I})} \times (n^3 + pw(\mathcal{I})^3 \times 2^{2pw(\mathcal{I})}))$  and  
 280 thus  $\mathcal{O}(n^4 \times 2^{4pw(\mathcal{I})})$ . The theorem follows.  $\square$

281 **Corollary 1.**  *$P|prec, p_i = 1, r_i, d_i|_\star$  is fixed-parameter tractable by the path-*  
 282 *width  $pw(\mathcal{I})$ .*

283 *Proof.* By Theorem 1, an instance  $\mathcal{I}$  has a solution if and only if there exists  
 284 a path in  $\mathcal{G}$  from a vertex of  $\mathcal{T}_1$  to one from  $\mathcal{T}_{k^*}$ . By removing vertices  
 285 without successor, it is equivalent to test that the remaining set  $|\mathcal{T}_1| > 0$ . By  
 286 Theorem 2, building  $\mathcal{G}$  is in time  $\mathcal{O}(n^4 2^{4pw(\mathcal{I})})$ , the corollary.  $\square$

287 Corollary 2 is a simple outcome of Theorem 2.

288 **Corollary 2.** *Let  $\mathcal{I}$  be an instance of  $P|prec, p_i = 1, r_i, d_i|*$  such that  $pw(\mathcal{I})$*   
289 *belongs to  $\mathcal{O}(\log_2 n)$ . The existence of a feasible schedule can be solved in*  
290 *time  $\mathcal{O}(n^8)$ .*

291 By augmenting our algorithm by a binary search on an upper bound of the  
292 objectives, we obtain in the next corollary the fixed-parameter tractability  
293 of two optimization problems:

294 **Corollary 3.** *Scheduling problems  $P|prec, p_i = 1, r_i|C_{\max}$  and  $P|prec, p_i =$*   
295  *$1, r_i|L_{\max}$  are fixed-parameter tractable by the pathwidth  $pw(\mathcal{I})$ .*

## 296 6. Conclusion and perspectives

297 For any instance  $\mathcal{I}$  of the decision problem  $P|prec, p_i = 1, r_i, d_i|*$ , we in-  
298 troduced the pathwidth  $pw(\mathcal{I})$  as a simple measure of the parallelism. We  
299 proved that the previous decision problem is fixed-parameter tractable by  
300 the parameter  $pw(\mathcal{I})$ . This result can be easily extended to  $P|prec, p_i =$   
301  $1, r_i|C_{\max}$  and  $P|prec, p_i = 1, r_i|L_{\max}$  using a binary search. As far as  
302 we know, it is the first fixed-parameter algorithm for these scheduling prob-  
303 lems. Extensions of this result should be considered to solve more general  
304 scheduling problems with precedence and resource constraints.

305 *Acknowledgments.* I thank Claire Hanen for many stimulating discussions  
306 and Theo Pedersen for his proofreading of the paper. I am also very grateful  
307 to the three reviewers for their helpful recommendations.

308 *Funding.* This research did not receive any specific grant from funding agen-  
309 cies in the public, commercial, or not-for-profit sectors.

## 310 References

- 311 [1] Baptiste, P., Timkovsky, V.G., 2004. Shortest path to nonpreemptive  
312 schedules of unit-time jobs on two identical parallel machines with min-  
313 imum total completion time. *Math. Methods Oper. Res.* 60, 145–153.
- 314 [2] van Bevern, R., Bredereck, R., Bulteau, L., Komusiewicz, C., Talmon,  
315 N., Woeginger, G.J., 2016. Precedence-constrained scheduling problems  
316 parameterized by partial order width, in: Kochetov, Y., Khachay, M.,

- 317 Beresnev, V., Nurminski, E., Pardalos, P. (Eds.), Discrete Optimization  
318 and Operations Research, Springer International Publishing. pp. 105–  
319 120.
- 320 [3] Bodlaender, H.L., 1992. A tourist guide through treewidth. *Acta Cy-*  
321 *bern.* 11, 1–21.
- 322 [4] Bodlaender, H.L., Fellows, M.R., 1995.  $W[2]$ -hardness of precedence  
323 constrained  $k$ -processor scheduling. *Oper. Res. Lett.* 18, 93–97.
- 324 [5] Coffman Jr., E.G., Dereniowski, D., Kubiak, W., 2012. An efficient  
325 algorithm for finding ideal schedules. *Acta Inf.* 49, 1–14.
- 326 [6] Cygan, M., Fomin, F.V., Kowalik, Ł., Lokshtanov, D., Marx, D.,  
327 Pilipczuk, M., Pilipczuk, M., Saurabh, S., 2015. *Parameterized Al-*  
328 *gorithms.* 1st ed., Springer Publishing Company, Incorporated.
- 329 [7] Dolev, D., Warmuth, M.K., 1984. Scheduling precedence graphs of  
330 bounded height. *J. Algorithms* 5, 48–59.
- 331 [8] Downey, R.G., Fellows, M.R., 2013. *Fundamentals of Parameterized*  
332 *Complexity.* 1st ed., Springer-Verlag London.
- 333 [9] Du, J., Leung, J.Y.T., Young, G.H., 1991. Scheduling chain-structured  
334 tasks to minimize makespan and mean flow time. *Information and Com-*  
335 *putation* 92, 219 – 236.
- 336 [10] Garey, M., Johnson, D., Tarjan, R., Yannakakis, M., 1983. Scheduling  
337 opposing forests. *SIAM Journal on Algebraic Discrete Methods* 4, 72–93.
- 338 [11] Graham, R., Lawler, E., Lenstra, J., Kan, A., 1979. Optimization and  
339 approximation in deterministic sequencing and scheduling: a survey, in:  
340 Hammer, P., Johnson, E., Korte, B. (Eds.), *Discrete Optimization II.*  
341 Elsevier. volume 5 of *Annals of Discrete Mathematics*, pp. 287 – 326.
- 342 [12] Günther, E., König, F.G., Megow, N., 2014. Scheduling and packing  
343 malleable and parallel tasks with precedence constraints of bounded  
344 width. *Journal of Combinatorial Optimization* 27, 164–181.
- 345 [13] Mnich, M., van Bevern, R., 2018. Parameterized complexity of machine  
346 scheduling: 15 open problems. *Computers and Operations Research* 100,  
347 254 – 261.

- 348 [14] Möhring, R.H., 1989. Computationally tractable classes of ordered sets,  
349 in: Rival, I. (Ed.), Algorithms and Order. Springer Netherlands, Dor-  
350 drecht, pp. 105–193.
- 351 [15] Prot, D., Bellenguez-Morineau, O., 2018. A survey on how the structure  
352 of precedence constraints may change the complexity class of scheduling  
353 problems. *Journal of Scheduling* 21, 3–16.