Minimizing the makespan for a UET bipartite graph on a single processor with an integer precedence delay

Alix MUNIER KORDON

Laboratoire LIP6,

4 place Jussieu, 75 252 Paris cedex 05

Alix.Munier@lip6.fr

Abstract

We consider a set of tasks of unit execution times and a bipartite precedence delays graph with a positive precedence delay d : an arc (i, j) of this graph means that j can be executed at least d time units after the completion time of i. The problem is to sequence the tasks in order to minimize the makespan.

Firstly, we prove that the associated decision problem is NP-complete. Then, we provide a non trivial polynomial time algorithm if the degree of every tasks from one of the two sets is 2. Lastly, we give an approximation algorithm with ratio $\frac{3}{2}$.

1 Introduction

Single and multiprocessors scheduling problems have been extensively studied in the literature [16]. Scheduling problems with precedence delays arise independently in several important applications and many theoretical studies were devoted to these problems : this class of problems was considered for resource-constrained scheduling problem [3, 13]. It was also studied as a relaxation for the job-shop problem [1, 8]. For computer systems, it corresponds to the basic pipelines scheduling problems [15, 20].

An instance of a scheduling problem with precedence delays is usually defined by a set of tasks $T = \{1, \ldots, n\}$ with durations $p_i, i \in T$, an oriented precedence graph $G = (T, E)$ and integer delays $d_{ij} \geq 0$, $(i, j) \in E$. For

Table 1: Complexity results

NP-Hard Problem	Reference
$1\|\text{chains}, d_{ij} = d\|C_{\max}$	Wikum et al.[23]
$1\|\text{prec}, d_{ij} = d, p_i = 1\|C_{\max}$	Leung et al.[18]

every arc $(i, j) \in E$, task j can be executed at least d_{ij} time units after the completion time of i. The number of processors is limited. The problem is to find a schedule minimizing the makespan, or other regular criteria. Using standard notations [16], the minimization of the makespan is denoted by $\mathrm{P}|\text{prec}, d_{ij}|C_{\max}$.

In this paper, we suppose that the graph G is bipartite : T is split into two sets X and Y and every arc $(i, j) \in E$ verifies $i \in X$ and $j \in Y$. We also consider that there is only one processor, the duration of tasks is one and that the delay is the same for every arc. This problem is noted $1|\text{bipartite}, d_{ij} = d, p_i = 1|C_{max}$. The decision problem associated is called SEQUENCING WITH DELAYS and is defined as :

- Instance : A bipartite oriented graph $G = (X \cup Y, E)$, a positive delay d and a deadline D.
- Question : is there a solution to the sequencing problem with a delay d and a makespan smaller than or equal to D?

We prove in section 2 that $1|\text{bipartite}, d_{ij} = d, p_i = 1|C_{max}$ is NP-Hard. The complexity of this problem was a challenging question since several authors proved the NP-Hardness of more general instances of this problem as shown in the table 1. In section 3, we prove that if the degree of every task in X is 2, then the problem is polynomial and we provide a greedy algorithm to solve it.

Several authors have adapted the classical polynomial algorithms for m processors and particular graphs structures to a sequencing problem with a unique delay as shown in the table 2. Note that Bampis [2] proved that $\mathrm{P}|\text{bipartite}, p_i = 1|C_{max}$ is NP-Hard, but his transformation doesn't seem to be easily extended to our problem.

Wikum et al. [23] also proved several complexity results, polynomial special cases and approximation algorithms for unusual particular classes of graphs (in fact, subclasses of trees). Munier and Sourd proved that $1|\text{chains}, d_{ij} = d, p_i = p|C_{\max}$ is polynomial. Lastly, Engels et al.[9] have

Table 2: Polynomial special cases

Polynomial Problem	Reference	Comments
$1\|\text{tree}, d_{ij} = d, p_i = 1\|C_{max}$	Bruno et al.[6]	Based on [14]
$1\|\text{prec}, d_{ij} = 1, p_i = 1\|C_{max}$	Leung et al. [18]	Based on [7]
$1\|\text{interval orders}, d_{ij} = d, p_i = 1\|C_{max}$	Leung et al.[18]	Based on [21]

developed a polynomial algorithm for $P|\text{tree}, d_{ij} \leq D, p_i = 1|C_{\max}$ if D is a constant value.

At last, there are some approximation algorithms for problems with delays : Graham's list scheduling algorithm [11] was extended to $P|\text{prec. delays}, d_{ij} = k, p_j = 1|C_{\max}$ to give a worst-case performance ratio of $2 - 1/(m(k+1))$ [15, 20]. This result was extended by Munier et al. [19] to $P|\text{prec. delays}, d_{ij}|C_{\max}$. Bernstein and Gerner [5] study the performance ratio of the Coffman-Graham algorithm for $P|\text{prec. delays}, d_{ij} = d, p_i = 1|C_{\max}$ and slightly improve it in [4]. Schuurman [22] developed a polynomial approximation scheme for a particular class of precedence constraints. We prove in section 4 that the bound 2 of Graham's list algorithm may be achieved in the worst case for $1|\text{bipartite}, d_{ij} = d, p_i = 1|C_{max}$ and we develop a simple algorithm with worst case performance ratio equal to 3/2 for this problem.

2 Complexity of the problem

Let us consider a non oriented graph $G = (V, E)$ and an ordering L of the vertices of G (*ie*, a one-to-one function $L : V \to \{1, \ldots, |V|\}$). For all integer $i \in \{1, \ldots, |V|\}$, the set $V_L(i) \subset V$ is :

$$V_L(i) = \{v \in V, L(v) \leq i \text{ and } \exists u \in V, \{v, u\} \in E \text{ and } L(u) > i\}$$

VERTEX SEPARATION is then defined as :

- Instance : A non oriented graph $G = (V, E)$ and a positive integer K.
- Question : Is there an ordering L of the vertices of G such that, for all $i \in \{1, \ldots, |V|\}$, $|V_L(i)| \leq K$?

This problem is proved to be NP-complete in [17]. For the following, our proofs will be more elegant if we consider the converse ordering of the tasks. Let $n = |V|$. If we set, $\forall v \in V$, $L'(v) = n - L(v)$, $j = n - i + 1$ and $B_{L'}(j) = V_L(i)$, we get for every value $j \in \{1, \ldots, n\}$:

$$B_{L'}(j) = \{v \in V, L'(v) > j \text{ and } \exists u \in V, \{v, u\} \in E \text{ and } L'(u) \leq j\}$$

So, the equivalent INVERSE VERTEX SEPARATION problem may be defined as :

- Instance : A non oriented graph $G = (V, E)$ and a positive integer K.
- Question : Is there an ordering L of the vertices of G such that, for all $i \in \{1, \ldots, |V|\}$, $|B_L(i)| \leq K$?

We prove the following theorem :

Theorem 2.1. *There exists a polynomial transformation f from INVERSE VERTEX SEPARATION to SEQUENCING WITH DELAYS.*

Proof. Let I be an instance of INVERSE VERTEX SEPARATION. The associated instance $f(I)$ is given by a bipartite graph $G' = (X \cup Y, E')$, a delay d and a deadline D defined as :

1. To any vertex $v \in V$ is associated two elements $x_v \in X$ and $y_v \in Y$ and an arc $(x_v, y_v) \in E'$.
2. To any edge $\{u, v\} \in E$ is associated the arcs (x_u, y_v) and (x_v, y_u) in E'.
3. The delay is $d = n - 1 - K$ and the deadline $D = 2n$.

f can be clearly computed in polynomial time (see an example figure 1).

Let us suppose that L is a solution to the instance I. Then, we build a solution to $f(I)$ as follows :

1. Tasks from Y are executed between time n and $2n$ following L : they are executed from $y_{L^{-1}(1)}$ to $y_{L^{-1}(n)}$.
2. Let us define the partition $P_i, i = 1 \ldots n$ of X as :

$$P_i = \{x_{L^{-1}(i)}\} \cup \{x_u, u \in B_L(i)\} - \bigcup_{j=1}^{i-1} P_j$$

   Tasks from X are executed between 0 and n following $P_1 \ldots P_n$.

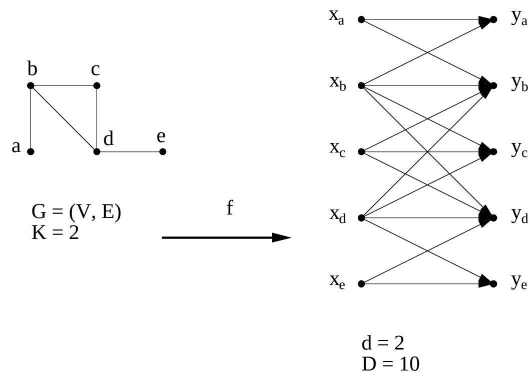

Figure 1: Example of transformation f

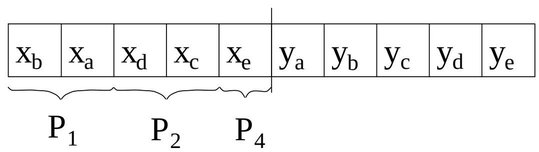

Figure 2: The schedule associated with L

For example, if we consider the order defined by $L(a) = 1$, $L(b) = 2$, $L(c) = 3$, $L(d) = 4$ and $L(e) = 5$, the sets P_i, $i = 1 \dots 5$, are defined by $P_1 = \{x_a, x_b\}$, $P_2 = \{x_c, x_d\}$, $P_3 = \emptyset$, $P_4 = \{x_e\}$ and $P_5 = \emptyset$. Figure 2 shows the corresponding solution for $f(I)$ for our example.

We have to prove now that this schedule fulfill all the precedence delays of G'. Let us consider the task $y_{L^{-1}(i)}, i = 1 \dots n$. We must show that all its predecessors in G' are completed at time $(n + i - 1) - d = K + i$.

1. We claim that all the predecessors of $y_{L^{-1}(i)}$ in G' are in $\bigcup_{j=1}^{i} P_j$. Indeed, $x_{L^{-1}(i)} \in P_j, j \leq i$ by construction.

   The other predecessors of $y_{L^{-1}(i)}$ are vertices x_v with v adjacent to $u = L^{-1}(i)$ in G. Now, if $L(v) < L(u)$, then $x_v \in P_k$ with $k \leq L(v)$. Otherwise, $v \in B_L(i)$ so $x_v \in P_k$ with $k \leq L(u)$.

2. We show that $|\bigcup_{j=1}^{i} P_j| \leq K + i$. Indeed, this set is composed by : [1] i tasks $x_{L^{-1}(j)}$, $j = 1 \dots i$, and [2] tasks x_u with $L(u) > i$, so $u \in B_L(i)$.

So, we built a solution to the instance $f(I)$.

Now, let us consider that we have a solution to $f(I)$. Since the graph G' is bipartite, we can exchange the tasks such that tasks from X are all completed before the first task from Y. We build an order L from tasks in Y such that, $\forall i \in \{1, \ldots, n\}$, $L^{-1}(i)$ is the task $u \in V$ such that y_u is executed at time $n + i - 1$. Then, we must prove that, $\forall i \in \{1, \ldots n\}$, $|B_L(i)| \leq K$.

Let consider $i \in \{1, \ldots, n\}$. Tasks executed during the interval $[0, K+i)$ can be decomposed into [1] $x_{L^{-1}(1)} \ldots x_{L^{-1}(i)}$ and [2] A set Q_i of K other tasks from $X \cup Y$.

Let be $v \in B_L(i)$. We claim that $x_v \in Q_i$. Indeed, we get that $L(v) > i$ and there exists $u \in V$ with $L(u) \leq i$ and $\{u, v\} \in E$. By definition of G', we have then $(x_v, y_u) \in E$, so $x_v \in Q_i$.

We deduce that $|B_L(i)| \leq |Q_i| = K$. □

Corollary 2.2. $1|bipartite, d_{ij} = d, p_i = 1|C_{max}$ *is NP-Hard.*

3 A polynomial special case

Let us consider a non oriented connected graph $G = (V, E)$ without loops (*i.e.* without edges $\{u, u\}$, $u \in V$) and an ordering L of the vertices. We set $|V| = n$. $\forall i \in \{1, \ldots, n\}$, we define the sequences $E_L(i)$ by :

$$E_L(i) = \{\{u, v\} \in E,\ L(u) \leq i\}$$

$E_L(i)$ is the set of edges adjacent to at least one vertices in $\{L^{-1}(1), \ldots, L^{-1}(i)\}$.

We define the problem MIN ADJACENT SET LINEAR ORDERING by :

- Instance : A non oriented graph $G = (V, E)$ without loops and a positive integer K.
- Question : Is there an ordering L of the vertices of G such that, for all $i \in \{1, \ldots, |V|\}$, $|E_L(i)| \leq K + i$?

Notice that the formulation of this problem is quite similar to MIN-CUT LINEAR ARRANGEMENT [10], which is NP-complete. In the following, we consider the subproblem Π of SEQUENCING WITH DELAYS with the restriction that the degree of every vertex from X is exactly 2.

Theorem 3.1. *There exists a polynomial transformation from* Π *to MIN ADJACENT SET LINEAR ORDERING*

Proof. Let us consider an instance I of Π given by a bipartite graph $G = (X \cup Y, E)$, a delay d and a deadline D. We build an instance $f(I)$ of MIN ADJACENT SET LINEAR ORDERING as follows :

- $G' = (Y, E')$. For every $x \in X$ with (x, y_1) and $(x, y_2) \in E$ is associated an edge $e_x = \{y_1, y_2\}$ in E'.
- the value $K = D - d - |Y| - 1$.

f can be computed in polynomial time. We prove now that f is a polynomial transformation (see figure 3 for an example)

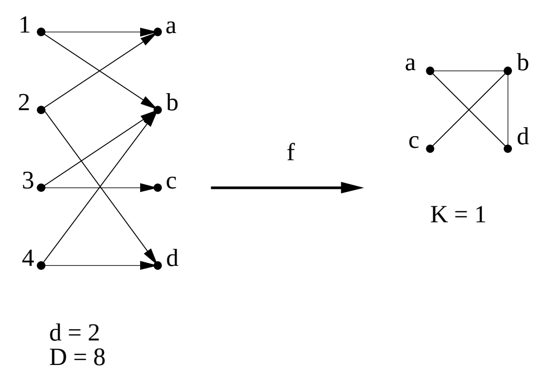

Figure 3: Example of transformation f

Let us suppose that a solution to I is given. Then, without loosing generality, we can suppose that the tasks from X are performed during $[0, \ldots, |X|)$ and tasks from Y during $[D - |Y|, \ldots, D)$. We build a linear ordering L following the sequencing order of tasks Y : $\forall i \in \{1, \ldots, |Y|\}$, $L(i)$ is the ith task of Y in the schedule.

$\forall i \in \{1, \ldots, |Y|\}$, let be $t = D - |Y| + (i - 1) = K + i + d$ the starting time of the task $L^{-1}(i)$ from Y. At time $t - d = K + i$, all the predecessors of $L^{-1}(1), \ldots, L^{-1}(i)$ must be completed. Now, for every edge $e_x \in E_L(i)$ is associated exactly one of those predecessors. So, $|E_L(i)| \leq K + i$.

Conversely, let us suppose that a solution to $f(\Pi)$ is given. Then, we perform tasks from Y following L during the interval $[D - |Y|, \ldots, D)$. We define then the following sequence $X_i \subset X$:

1. $X_1 = \{x \in X, e_x \in E_L(1)\}$,
2. $\forall i = 2, \ldots, n$, $X_i = \{x \in X, e_x \in E_L(i)\} - \bigcup_{j=1}^{i} X_j$.

Notice that, by construction that, $\forall i \in \{1, \dots, n\}, \bigcup_{j=1}^{i} X_i = \{e_x \in E_L(i)\}$. Tasks of X are performed during $[0, \dots, |X)$ following $X_1, X_2 \dots X_n$. Every task from $\bigcup_{j=1}^{i} X_i$ is then completed at time $K + i$ (see figure 4 for the corresponding schedule).

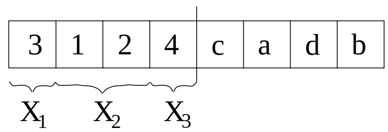

Figure 4: A corresponding schedule

We must prove that the delays constraints are fulfilled : let us consider the task $y = (L^{-1}(i))$. For every task $x \in \Gamma^{-1}(y)$ is associated $e_x \in E_L(i)$. So, $x \in \bigcup_{j=1}^{i} X_i$ and is completed at time $K + i$. Since y is performed at time $t = D - |Y| + i - 1$, we get :

$$t - (K + i) = D - |Y| + i - 1 - (K + i) = d$$

So, the delays are fulfilled. □

Theorem 3.2. *Let us consider an instance I of MIN ADJACENT SET LINEAR ORDERING given by a graph $G = (V, E)$ and an integer $K > 0$. A necessary and sufficient condition for the existence of a solution is that*

$$|E| \leq K + |V| - 1$$

Proof. The condition is necessary : since the graph G is connected without loops, every linear ordering L verifies $E_L(n - 1) = E$. So, if L verifies the condition, we get the condition of the theorem.

The condition is sufficient : let us consider a linear ordering L and a family of graph G_i, $i = 0, \dots, n$ defined such that,

- $G_0 = G$,
- $\forall i = 1, \dots, n$, we choose a vertex u in the subgraph $G_{i-1} = (V - \{L^{-1}(1), \dots, L^{-1}(i-1)\}, E)$ with a minimum degree in G_{i-1} and we set $L(u) = i$.
- $G_n = \emptyset$.

We note E_i the edges of G_i. Notice that, $\forall i = 1, \dots, n$, the two sets $E_L(i)$ and E_i are a partition of E.

We prove by contradiction that the linear ordering L is a solution to MIN ADJACENT SET LINEAR ORDERING.

- Let us suppose that $|E_L(1)| \geq K+2$, then the degree of any vertex in G is greater than or equal to $K+2$. So, $2|E| \geq |V|(K+2)$. By hypothesis, we get $2K+2|V|-2 \geq K|V|+2|V|$, so $K(2-|V|) \geq 2$.

  Since $K > 0$, we get that $|V| < 2$, so $|V| = 1$. In this case, we get $|E_L(1)| = |E| = 0$, which contradicts $|E| \geq K+2$.

- Now, let us suppose that, for $i < n-2$, $\forall j \in \{1,\ldots,i\}$, $|E_L(j)| \leq K+j$ and that $|E_L(i+1)| \geq (i+1)+K+1$. For every vertex $u \in G_i$, we set $d_{G_i}(u)$ the degree of u in G_i.

  The total number of edges verifies $|E| = |E_L(i+1)| + |E_{i+1}|$.

  1. By hypothesis, $|E_L(i+1)| \geq (i+1)+K+1$.
  2. By definition of the sequences G_i, $|E_{i+1}| = |E_i| - d_{G_i}(L^{-1}(i+1))$. Since $u = L^{-1}(i+1)$ is the vertex of G_i with a minimum degree, the number of arcs of G_i verifies

     $$2|E_i| \geq (n-i)d_{G_i}(L^{-1}(i+1))$$

     So,

     $$|E_{i+1}| \geq \frac{1}{2}(n-i)d_{G_i}(L^{-1}(i+1)) - d_{G_i}(L^{-1}(i+1))$$

     We show that $d_{G_i}(L^{-1}(i+1)) \geq 2$. Indeed, let us denote by $e(k) = \{L^{-1}(i+1), L^{-1}(k)\}$ an edge of G adjacent to $L^{-1}(i+1)$. Then, we get easily that $E_L(i+1) - E_L(i) = \{e(k) \in G_i\}$, so

     $$d_{G_i}(L^{-1}(i+1)) = |E_L(i+1)| - |E_L(i)| \geq (i+1)+K+1-(K+i) = 2$$

     We deduce that

     $$|E_{i+1}| \geq \frac{n-i-2}{2}d_{G_i}(L^{-1}(i+1)) \geq n-i-2$$

  So, the total number of edges of G verifies :

  $$|E| = |E_L(i+1)| + |E_{i+1}| \geq (i+1)+K+1+n-i-2 = |V|+K$$

  which contradicts the hypothesis of the theorem. □

Notice that this proof is constructive : if the condition of the theorem is fulfilled, one can easily implements a greedy polynomial algorithm to build a linear ordering.

Corollary 3.3. Π *is polynomial.*

If we heavily sort the the vertices at each step of the algorithm, the complexity of the algorithm will be bounded by $O(n^2 \log n + m)$.

4 An Approximation algorithm

In this section, we consider the analysis of the performances of two approximation algorithms.

The first one is the classical Graham list scheduling algorithm [12]. At each time t, a schedulable task is chosen to be performed without any priority rule. For the bipartite graph $G = (X \cup Y, E)$, it consists on performing tasks from X in any order and tasks from Y as soon as possible. Several authors show that the performance ratio of this algorithm is upper bounded asymptotically by 2 [15, 20, 19]. We prove here that this bound is reached for bipartite graphs :

Theorem 4.1. *The performance ratio of a list scheduling for a bipartite graph tends asymptotically to* 2*.*

Proof. Let us consider a value $d > 0$ and a bipartite graph $G = (X \cup Y, E)$ with $X = \{a_1, \dots, a_d\} \cup \{b\}$, $Y = \{c\}$ and $E = \{(b, c)\}$. In the worst case for the Graham list scheduling algorithm, tasks $\{a_1, \dots, a_d\}$ are performed first. We get then a schedule of length $l_1 = 2d + 2$.

Now, we can get a schedule without idle slots if we perform b first. The length of this second schedule is then $l_2 = d + 2$.

The performance ratio is then bounded by : $r = \frac{2d+2}{d+2} = 2 - \frac{2}{d+2} \rightarrow_{d \rightarrow \infty}$ 2. □

We present now a slightly better approximation algorithm : let us suppose that $G = (X \cup Y, E)$ with $|X| = n$, $|Y| = m$ and $n \geq m$. In the opposite, we modify the orientation of the edges and we consider the graph $G' = (Y \cup X, E')$. We can get a feasible schedule for G by considering the inverse order of a schedule for G'.

Let us consider the set X_1 of tasks from X with a strictly positive outdegree (*i.e.*, X_1 is the set of X with at least one successor in Y). The idea is to apply a list scheduling algorithm which performs tasks from X_1 before those from $X_2 = X - X_1$.

We denote by C_{opt} (resp. C_H) the makespan of an optimal schedule (resp. a schedule obtained using this algorithm). We set $|X_i| = n_i, i = 1, 2$ and $p = \max(0, d + 1 - n_2 - m)$. We prove the following upper bound on C_{opt} :

Lemma 4.2. $C_{opt} \geq n + m + p$*.*

Proof. The last task of X_1 is performed at time $t \geq n_1$ and has at least one successor in Y, so $C_{opt} \geq n_1 + d + 1$. Now, if $p = d + 1 - n_2 - m$,

$n+m+p = n+m+d+1-n_2-m = n_1+d+1$ and the inequality is true. Otherwise, $p=0$ and we get obviously $C_{opt} \geq n+m$. □

Theorem 4.3. *The performance ratio of this algorithm is bounded by $\frac{3}{2}$.*

Proof. We denote by $\mathcal{I}$ the idle slots of the schedule obtained by our algorithm. We get, using the previous lemma :

$$C_H = n+m+|\mathcal{I}| \leq C_{opt} + (|\mathcal{I}| - p)$$

1. If $|\mathcal{I}| \leq p$, we get the theorem.

2. Let us assume now that $|\mathcal{I}| > p$. We build a subset $\mathcal{I}_p \subset \mathcal{I}$ by removing from $\mathcal{I}$ the pth first idle slots in our schedule. Let be an element $k \in \mathcal{I}_p$ and $t(k)$ the time of this idle slot.

   Clearly, by definition of $\mathcal{I}_p$, $t(k) \geq p+n$. Moreover, there is at least one task from $y \in Y$ performed after $t(k)$ such that y is not ready at time $t(k)$, so $t(k) \leq n_1 + d$. We get

   $$|\mathcal{I}| - p = |\mathcal{I}_p| \leq n_1 + d - (p+n)$$

   Then,

   $$|\mathcal{I}| - p = |\mathcal{I}_p| \leq d - n_2 - \max(0, d+1-n_2-m)$$

   We deduce that

   $$|\mathcal{I}_p| \leq \min(d-n_2, m-1)$$

   So, $|\mathcal{I}_p| \leq |Y|$.

   Now, the inequality between C_H and C_{opt} becomes :

   $$C_H \leq C_{opt} + |\mathcal{I}_p| \leq C_{opt} + |Y|$$

   Since $|Y| \leq |X|$, we get that $|Y| \leq \frac{1}{2}(|X|+|Y|) \leq \frac{1}{2}C_{opt}$ and we get the theorem. □

We can prove that the bound $\frac{3}{2}$ is asymptotically tight : indeed, let us consider an integer $n > 0$ and the bipartite graph $G = (X \cup Y, E)$ with $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_n\}$ and the arcs $E = \{(x_i, y_j), 1 \leq j \leq i \leq n\}$. We set $d = n-1$. Note that $|X| = n = |Y|$.

If we perform task from X such that $t(x_i) = i-1, i = 1, \ldots, n$, then tasks from Y can't be performed before $n+d-1$. So, we get a makespan $L_1 = 3n-2$.

Now, if we perform task from from X such that $t(x_i) = n-i, i = 1, \ldots, n$, then we get a schedule without idle slots with makespan $L_2 = 2n$.

So, we get $\frac{L_1}{L_2} \rightarrow_{n \rightarrow +\infty} \frac{3}{2}$.

5 Conclusions

Several new questions arise from the results presented here :

- In order to study the borderline between NP-complete and polynomial problems, the complexity of the problem with a bipartite graph where the degree of vertices from X does not exceed 3 is an interesting problem.
- The existence of better approximation algorithms is also an interesting question.

References

[1] E. Balas, J. K. Lenstra, and A. Vazacopoulos. The one machine problem with delayed precedence constraints and its use in job-shop scheduling. *Management Science*, 41:94 – 109, 1995.

[2] E. Bampis. The complexity of short schedules for uet bipartite graphs. *RAIRO Operations Research*, 33:367 – 370, 1999.

[3] M. Bartusch, R. H. Möhring, and F. J. Radermacher. Scheduling project networks with resource constraints and time windows. *Annals of Operations Research*, 16:201 – 240, 1988.

[4] D. Bernstein. An improved approximation algorithm for scheduling pipelined machines. In *Int. Conf. on Parallel Processing*, volume 1, pages 430 – 433, 1988.

[5] D. Bernstein and I. Gertner. Scheduling expressions on a pipelined processor with a maximum delay of one cycle. *ACM Transactions on Programming Languages and Systems*, 11:57 – 66, 1989.

[6] J. Bruno, J. W. Jones, and K. So. Deterministic scheduling with pipelined processors. *IEEE Transactions on Computers*, C-29:308 – 316, 1980.

[7] E.G. Coffman and R.L. Graham. Optimal scheduling for two processor systems. *Acta Informatica*, 1:200 – 213, 1972.

[8] S. Dauzère-Peres and J.-B. Lasserre. A modified shifting bottleneck procedure for job–shop scheduling. *Int. J. Production Research*, 31:923 – 932, 1993.

[9] D.W Engels, J. Feldman, D.R Karger, and M. Ruhl. Parallel processor scheduling with delay constraints. Rapport Interne MIT.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP–Completeness*. Freeman, San Francisco, 1979.

[11] R. L. Graham. Bounds for certain multiprocessing anomalies. *Bell System Tech. J.*, 45:1563 – 1581, 1966.

[12] R. L. Graham. Bounds on the performance of scheduling algorithms. In E. G. Coffman, editor, *Computer and Job-shop Scheduling Theory*. John Wiley Ltd., 1976.

[13] W. Herroelen and E. Demeulemeester. Recent advances in branch-and-bound procedures for resource-constrained project scheduling problems. In P. Chrétienne, E. G. Coffman Jr., J. K. Lenstra, and Z. Liu, editors, *Scheduling Theory and its Applications*, chapter 12, pages 259 – 276. John Wiley & Sons, 1995.

[14] T.C. Hu. Parallel sequencing and assembly lines problems. *Operations Research*, 9:841 – 848, 1961.

[15] E. Lawler, J. K. Lenstra, C. Martel, B. Simons, and L. Stockmeyer. Pipeline scheduling: A survey. Technical Report RJ 5738 (57717), IBM Research Division, San Jose, California, 1987.

[16] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. Sequencing and scheduling: Algorithms and complexity. In S. C. Graves, A. H. G. Rinnooy Kan, and P. H. Zipkin, editors, *Logistics of Production and Inventory*, volume 4 of *Handbooks in Operations Research and Management Science*, chapter 9, pages 445 – 522. North–Holland, Amsterdam, The Netherlands, 1993.

[17] T. Lengauer. Black-white pebbles and graph separation. *Acta Informatica*, 16:465 – 475, 1981.

[18] J. Y.–T. Leung, 0. Vornberger, and J. Witthoff. On some variants of the bandwidth minimization problem. *SIAM J. Computing*, 13:650 – 667, 1984.

[19] A. Munier, M. Queyranne, and A. S. Schulz. Approximation bounds for a general class of precedence constrained parallel machine scheduling problems. Number 1412 in Lecture Notes in Computer Science, pages

367 – 382. Springer, 1998. Proceedings of the 6th International IPCO Conference.

[20] K. W. Palem and B. Simons. Scheduling time critical instructions on risc machines. In *Proceedings of the 17th Annual Symposium on Principles of Programming Languages*, pages 270 – 280, 1990.

[21] C.H Papadimitriou and M. Yannakakis. Schedulin interval-ordered tasks. *SIAM Journal on Computing*, 8(3), 1979.

[22] P. Schuurman. A fully polynomial approximation scheme for a scheduling problem with intree-type precedence delays. *Operations Research Letters*, 23:9 – 11, 1998.

[23] E. D. Wikum, D. C. Llewellyn, and G. L. Nemhauser. One–machine generalized precedence constrained scheduling problems. *Operations Research Letters*, 16:87 – 89, 1994.