Periodic Schedules For Linear Precedence Constraints

Claire Hanen and Alix Munier Kordon

Laboratoire LIP6

4 place Jussieu 75 252 Paris cedex 05

{*Claire.Hanen, Alix.Munier*}@*lip6.fr*

Abstract

We consider the computation of periodic cyclic schedules for linear precedence constraints graph: a linear precedence constraint is defined between two tasks and induces an infinite set of usual precedence constraints between their executions such the the difference of iterations is a linear function.The objective function is the minimization of the maximal period of a task. Firstly, we recall that this problem can be modelled using linear programming. Then, we develop a polynomial algorithm to solve it for unitary graphs, which is a particular class of linear precedence graph.We also show that a periodic schedule may not exists for this special case. In the general case, we compute a decomposition of the graph into unitary components and we suppose that a periodic schedule exists for each of them. We compute lower bounds on the periods and we show that an optimal periodic schedule may not achieve them. Then, we introduce the notion of quasi-periodic schedule, and we prove that this new class of schedule always reach these bounds.

Keywords: Periodic schedules, cyclic scheduling, linear precedence.

1 Introduction

Cyclic scheduling, in which a set of tasks has to be repeated infinitely, has been studied for several years and has yield many results [7, 6, 11]. The main practical applications are mass production in manufacturing systems as well as computing loops on parallel or pipelined processors or the synthesis of embedded systems.

We consider a set of tasks $\mathcal{T} = \{1, ..., n\}$. Let us denote by $< i, k >$ the k^{th} occurrence of a task $i \in \mathcal{T}$. A schedule σ assigns to each occurrence k of any task i a starting time $t^\sigma(< i, k >)$. We call average cycle time of a task i, the average time interval between two successive occurrences of i. It can be formally defined as:

$$limsup_{k \to \infty} \frac{t^\sigma(< i, k >)}{k}$$

Usually, the maximum average cycle time is to be minimized.

Several kinds of relations between the executions of the tasks may be defined. One of them, called uniform precedences [1], is the most usual way to extend precedence constraints to cyclic scheduling problems. A set of uniform precedence is usually represented as a multi-graph, called uniform graph, in which each node corresponds to a task, and each arc a from node $b(a) = i$ to node $e(a) = j$ has two valuations: its length l_a equals the processing time of task i, and its height h_a. An arc induces an infinite number of usual precedences as follows:

$$\forall k \geq 1, \quad < i, k > \quad precedes \quad < j, k + h_a >$$

The simplest way to execute the tasks is to build a periodic schedule. In this case, tasks starting times follow $t^\sigma(< i, k >) = t^\sigma(< i, 1 >) + w(k-1)$, where w is the period of the schedule.

It is well known [7, 10] that there exists a schedule satisfying the constraints induced by a graph if and only if there exists a periodic schedule. Moreover, the periodic schedule with the least period has the same average cycle time than the earliest schedule [3], and thus is optimal. It can be computed polynomially by shortest-path like algorithms.

This first model has been extended by Munier to linear precedence constraints. This extension is useful to model problems issued from the computation of loops on parallel processors as well as assembly lines [12, 7]. Notice that this model is slightly more general than generalized timed event graphs and synchronous dataflow [9]. A set of linear precedence constraints is usually expressed as a multi-graph G, the nodes of which represents the tasks, and the arcs of which the linear precedences. Each arc a from node $b(a) = i$ to node $e(a) = j$ has five integer values: $l_a \geq 1$ (processing time of task i), $p_a, p'_a \geq 1$, $q_a, q'_a \in \mathbb{Z}$.

An arc induces an infinite number of usual precedences as follows:

$$\forall k \geq 0, \quad < i, p_a k + q_a > \quad precedes \quad < j, p'_a k + q'_a >$$

Let us denote by $\pi_a = \dfrac{p'_a}{p_a}$ the weight of an arc a, and let us define the weight $\Pi(\mu)$ of a path μ as the product of the weight of its arcs. It has been proven in [13] that if there exists a schedule, all cycles of G have a weight not less than 1 and that this condition is not sufficient.

In the particular case of unitary graphs, in which the weight of any cycle equals 1, it has been shown that an equivalent uniform graph can be built, so that the existence of schedules can be checked on this graph as well as the construction of efficient schedules, using results on cyclic scheduling with uniform constraints. Unfortunately, the size of this new graph is not polynomial, so that all these conditions and algorithms are not polynomial.

In this paper, we tackle the problem of the construction of periodic schedules, in which each task has its own periodicity, which appear to be easy to implement in applications for embedded [4, 8], or production systems [2, 14]. In Section 2, we first recall that the problem can be modelled using linear programming: this modelling was first introduced by [5] for this particular problem. We also show that a graph G may not have a periodic schedule, even if an earliest schedule exists.

In Section 3, we study the particular case of unitary graphs, $i.e$ a strongly connected graph such that every circuit has a weight equal to 1.We show that the feasible periods of the tasks are proportional to a particular vector, and we deduce a simple polynomial algorithm to compute an optimal solution.

Section 4 is dedicated to the general case. We first compute lower bounds on optimal periods based on the decomposition of the graph into unitary components, and we show that this value may not be achieved by a periodic schedule. Then, we introduce quasi-periodic schedules, and we prove that if the graph is feasible, and if its unitary components can be scheduled periodically, we can build a quasi-periodic schedule with periods equal to these lower bounds.

2 Modelling the problem

Let us consider a linear precedence multi-graph $G = (\mathcal{T}, A)$. Notice that, unlike uniform constraints, linear constraint induces that in an optimal schedule (such as the earliest schedule), the tasks may have different average cycle times. For example, let us consider the linear precedence multigraph pictured by Figure 1. If we consider the arc $a = (2, 1)$, the corresponding precedence constraint is

$$\forall k \geq 0, \quad t(<2, k+1>) + 5 \leq t(1, 3k+1)$$

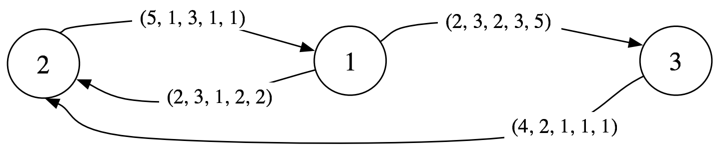

Figure 1: A linear precedence multi-graph G. Every arc a is labeled with $(l_{e(a)}, p_a, p'_a, q_a, q'_a)$

Intuitively, this constraint induces that the average cycle time of the task 2 is at most 3 times the average cycle time of the task 1. So, we extend the notion of periodic schedules to linear precedence constraints by allowing the tasks to have different periods.

Definition 2.1. Let us consider a linear precedence multi-graph $G = (\mathcal{T}, A)$. A schedule σ is said to be periodic if there exists two vectors of positive numbers $w = (w_1, \dots, w_n)$ and $t = (t_1, \dots, t_n)$, such that:

$$\forall k \geq 1, t^\sigma(< i, k >) = t_i + (k-1)w_i$$

Let us consider an arc $a \in A$. If σ is a periodic schedule, then

$$t^\sigma(< b(a), p_a k + q_a >) = t_{b(a)} + (p_a k + q_a - 1)w_{b(a)}$$
$$t^\sigma(< e(a), p'_a k + q'_a >) = t_{e(a)} + (p'_a k + q'_a - 1)w_{e(a)}$$

From that we derive easily the following lemma :

Lemma 2.2. *A periodic schedule meets the linear precedence constraints if and only if for any arc $a \in A$,*

$$\forall k \geq 0, t_{e(a)} - t_{b(a)} \geq l_a + (w_{b(a)}p_a - w_{e(a)}p'_a)k + w_{b(a)}(q_a - 1) - w_{e(a)}(q'_a - 1)$$

Since this inequality must be true $\forall k$, we must have $w_{b(a)}p_a - w_{e(a)}p'_a \leq 0$. We also notice that the right term of this inequality is maximal for $k = 0$. Moreover, the aim is to minimize the largest period of a task. So, the computation of an optimal periodic schedule for a linear precedence graph may be modelled by the following linear program:

$$\left\{\begin{array}{ll} & Min\ B \\ \forall\ a \in A & w_{b(a)}p_a - w_{e(a)}p'_a \leq 0 \\ & t_{e(a)} - t_{b(a)} \geq l_a + w_{b(a)}(q_a - 1) - w_{e(a)}(q'_a - 1) \\ \forall i \in \mathcal{T} & l_i \leq w_i \leq B \\ \forall i \in \mathcal{T} & 0 \leq t_i \end{array}\right. \quad (1)$$

For example, the system associated with the graph pictured by Figure 1 is:

$$\left\{\begin{array}{l} Min\ B \\ w_2 - 3w_1 = 0 \\ 3w_1 - 2w_3 \leq 0 \\ 2w_3 - w_2 \leq 0 \\ -5 \geq t_2 - t_1 \geq 2 + w_1 - w_2 \\ t_3 - t_1 \geq 2 + 2w_1 - 4w_3 \\ t_2 - t_3 \geq 4 \\ 2 \leq w_1 \leq B,\ 5 \leq w_2 \leq B,\ 4 \leq w_3 \leq B \\ t_1 \geq 0,\ t_2 \geq 0,\ t_3 \geq 0 \end{array}\right. \quad (2)$$

In the following sections, we will study the solutions of this linear program for unitary graphs and in the general case.

3 Unitary graphs

We assume here that G is a unitary graph, *i.e.* G is strongly connected and every cycle c of G has a weight $\Pi(c) = 1$.We develop a simple polynomial time algorithm to solve the previous linear program associated with G.

Lemma 3.1. *If G is unitary, then $\forall a \in A$, $\dfrac{w_{b(a)}}{w_{e(a)}} = \pi_a$.*

Proof. For any arc $a \in A$, $\dfrac{w_{b(a)}}{w_{e(a)}} \leq \dfrac{p'_a}{p_a} = \pi_a$. So, if μ is a path from node i to node j, then we should have $\dfrac{w_i}{w_j} \leq \Pi(\mu)$. Let us consider now an arc

$a \in A$. As G is unitary, it is strongly connected and thus there is a path μ from $e(a)$ to $b(a)$. Hence, if there exists a periodic schedule, we should have $\dfrac{w_{b(a)}}{w_{e(a)}} \leq \pi_a$ and $\dfrac{w_{e(a)}}{w_{b(a)}} \leq \Pi(\mu)$. As G is unitary, $\pi_a.\Pi(\mu) = 1$, so that the second inequality becomes $\dfrac{w_{b(a)}}{w_{e(a)}} \geq \pi_a$. □

Let us consider the set of feasible periods

$$\mathcal{W} = \{w = (w_1, .., w_n) \in \mathbb{Q}^{+n} / \forall a \in A, \frac{w_{b(a)}}{w_{e(a)}} = \pi_a\}$$

Using the same arguments as in [13], we prove now that every element $w \in \mathcal{W}$ verifies $w_i = \lambda W_i$, where λ is a strictly positive rational number and the vector $(W_1, ..., W_n)$ is a particular element from $\mathcal{W}$.

As G is unitary, all paths from 1 to i have the same weight. Let ρ_i be the weight of any path from node 1 to node i. We can compute easily in polynomial time two integers α_i, β_i such that $\rho_i = \frac{\alpha_i}{\beta_i}$ and $gcd(\alpha_i, \beta_i) = 1$. Let us define $\beta = lcm(\beta_1, \ldots, \beta_n)$, $\alpha = lcm(\alpha_1, \ldots, \alpha_n)$.

Let us define $N_i = \beta\rho_i$, which will be referred as the minimum expansion number of i in the following. It is proved in [13] that a uniform graph $Exp(G)$ can be build in which each node i of G is duplicated N_i times, that is equivalent to G in terms of precedence constraints. Notice that the number of duplicates might not be polynomial with respect to the size of the data. For the previous example, we get $\rho_1 = 1$, $\rho_2 = \frac{1}{3}$, and $\rho_3 = \frac{2}{3}$.We obtain $\alpha = 2$, $\beta = 3$ and the number of duplicates are $N_1 = 3$, $N_2 = 1$ and $N_3 = 2$.

Lemma 3.2. *$W = (\frac{\alpha}{\rho_1}, ..., \frac{\alpha}{\rho_n}) \in \mathcal{W}$. Moreover, any other element from $\mathcal{W}$ is a multiple of W.*

Proof. Let $a \in A$. By definition of W, $\frac{W_{b(a)}}{W_{e(a)}} = \frac{\rho_{e(a)}}{\rho_{b(a)}}$. But $\rho_{e(a)} = \pi_a \rho_{b(a)}$, hence $W \in \mathcal{W}$. Let $w \in \mathcal{W}$ and a task $i \in \mathcal{T}$. We know that $\dfrac{W_1}{W_i} = \dfrac{w_1}{w_i} = \dfrac{\rho_i}{\rho_1}$, hence $\forall i, \dfrac{W_1}{w_1} = \dfrac{W_i}{w_i}$. □

Let us define, for any arc $a \in A$ its height $h_a = W_{e(a)}(q'_a - 1) - W_{b(a)}(q_a - 1)$. Let us define the height $H(\mu)$ (*resp.* the length $L(\mu)$) of a path μ to be the sum of the heights (*resp.* lengths) of its arcs. We can now state the existence condition:

Theorem 3.3. *There exists a periodic schedule of a unitary graph G if and only if all cycles of G have a positive height. Moreover a periodic schedule satisfies the following inequalities for any arc a of A:*

$$t_{e(a)} - t_{b(a)} \geq l_a - \lambda h_a$$

Proof. Let us suppose that there exists a periodic schedule for G. As the periods satisfy lemma 3.1, we should have for any arc a of G:

$$t_{e(a)} - t_{b(a)} \geq l_a + w_{b(a)}(q_a - 1) - w_{e(a)}(q'_a - 1) \tag{3}$$

Now, according to lemma 3.2, there exists λ such that $w_{b(a)} = \lambda W_{b(a)}$ and $w_{e(a)} = \lambda W_{e(a)}$. Hence, we get $t_{e(a)} - t_{b(a)} \geq l_a - \lambda h_a$.

If μ is a cycle of G, summing these inequalities along the arcs of the cycle will lead to $0 \geq L(\mu) - \lambda H(\mu)$. Hence if the height of the cycle is non positive, this inequality will be violated.

Conversely, let us suppose that every cycle μ has a positive height.Then, for large enough λ^*, we get $L(\mu) - \lambda^* H(\mu) \leq 0$, so that the graph G with arcs valued by $l_a - \lambda^* h_a$ has non positive cycles. Let t_i be the maximum value $L(\mu_i) - \lambda^* H(\mu_i)$ of a path μ_i from node 1 to node i. t_i satisfies the inequalities of lemma 2.2, so

$$t^\sigma(< i, k >) = t_i + (k-1)\lambda^* W_i$$

defines a periodic schedule. □

The condition of the previous theorem can be checked polynomially by a longest path algorithm, together with a depth first search on a subgraph. Notice that if G is uniform, this condition is exactly the feasibility condition of the task system. Unfortunately, it is not a necessary condition of existence of a schedule. Indeed, if we consider the unitary graph defined by two nodes $1, 2$ and two arcs $u = (1, 2)$ and $v = (2, 1)$ with the following values: $p_u = 3$, $p'_u = 2$, $q_u = 1$, $q'_u = 1$, $p_v = 2$, $p'_v = 3$, $q_v = 2$, $q'_v = 2$, we can check that there is no periodic schedule, although the task system is feasible. Indeed, $W_1 = 2, W_2 = 3$, so that $h(u) = 0, h(v) = -1$. But the infinite precedence graph of all occurrences of tasks, shown in Figure 2 does not have any circuit, so that the earliest schedule exists.

Let us now assume we are given a unitary graph for which a periodic schedule exists. We can build the periodic schedule with minimum periods, *i.e.* a periodic schedule with maximum average cycle time (among periodic

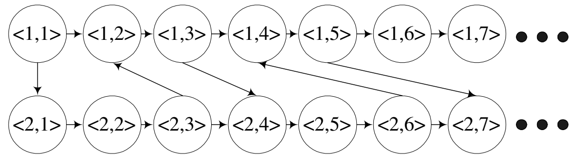

Figure 2: The developped graph of precedences

schedules). Indeed, all periods of tasks are multiple of a number λ. Hence minimizing λ, is equivalent to minimize all periods at the same time. Let

$$\lambda_{\min} = \max_{c \text{ } cycle \text{ } of G} \frac{L(c)}{H(c)}$$

Exactly as for uniform graphs, $\lambda_{\min}$ can be computed in polynomial time using a binary search combined with a longest path algorithm. For our previous example, we get the graph G valued by $l_a - \lambda h_a$ pictured by Figure 3, the vector $W = (2, 6, 3)$ and $\lambda_{\min} = \frac{7}{4}$.

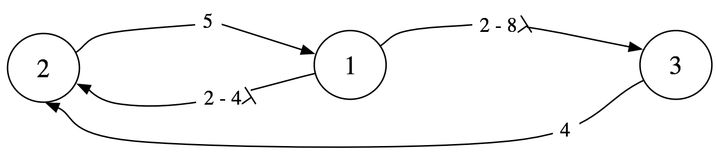

Figure 3: Graph G valued by $l_a - \lambda h_a$

Since in any feasible periodic schedule, the period of task i is not less than $\lambda_{min} W_i$, we get the following theorem:

Theorem 3.4. *Let us consider a unitary graph G such that a periodic schedule exists. For any $\lambda \geq \lambda_{\min}$, there exists a periodic schedule such that for any task i, the period i is $w_i^\lambda = \lambda W_i$. The starting time t_i of the first execution of i can be build in polynomial time, using a longest path algorithm. The optimal periodic schedule is defined for $\lambda = \lambda_{\min}$.*

4 General case

It is shown in [12] that every linear precedence graph $G = (\mathcal{T}, A)$ can be decomposed into unitary components $G^1, ..., G^r$ defined as follows:

Definition 4.1. For every linear precedence graph $G = (\mathcal{T}, A)$, there exists a unique partition of the set of nodes $\{\mathcal{T}^\alpha, \alpha = 1, ..., r\}$ such that:

1. $\forall \alpha \in \{1, ..., r\}$, a set of arcs $A^\alpha \subset A$ may be associated to $\mathcal{T}^\alpha$ such that the partial sub-graph $G^\alpha = (\mathcal{T}^\alpha, A^\alpha)$ is a unitary graph,

2. For every circuit c of G with weight $\Pi(c) = 1$, there exists $\alpha \in \{1, ..., r\}$ such that c is a circuit of G^α.

Notice that a periodic schedule of G induces a periodic schedule of each component G^α, $\alpha \in \{1, ..., r\}$. We thus assume that there exists a periodic schedule for each G^α, $\alpha \in \{1 \ldots, r\}$.

Lemma 4.2. *With any periodic schedule of G is associated a vector $\lambda^1, \ldots, \lambda^r$ such that $\forall \alpha \in \{1, \ldots, r\}$,*

1. $\lambda^\alpha \geq \lambda^\alpha_{\min}$

2. $\forall i \in V^\alpha$, $w_i = \lambda^\alpha W^\alpha_i$.

Proof. Let $\alpha \in \{1, \ldots, r\}$. According to lemma 3.2, we can define a vector W^α of $|\mathcal{T}^\alpha|$ values, such that in any periodic schedule of G^α all tasks $i \in \mathcal{T}^\alpha$ have a period $w_i = \lambda W^\alpha_i$ for some $\lambda \geq 0$. Hence in any periodic schedule of G, there exists a vector $\lambda^1, \ldots, \lambda^r$ such that the period of a task $i \in \mathcal{T}^\alpha$ satisfies $w_i = \lambda^\alpha W^\alpha_i$

Moreover, according to theorem 3.4, we can define $\forall \alpha \in \{1, \ldots, r\}$ a minimum value $\lambda^\alpha_{\min}$ such that $\lambda^\alpha \geq \lambda^\alpha_{\min}$. □

Now, let us consider an arc $a \in A$ which does not belong to a unitary component with $b(a) \in \mathcal{T}^\alpha$ and $e(a) \in \mathcal{T}^\beta$. By lemma 3.2, we must have $w_{b(a)}p_a - w_{e(a)}p'_a \leq 0$, so we get:

$$\lambda^\alpha W^\alpha_{b(a)} p_a - \lambda^\beta W^\beta_{e(a)} p'_a \leq 0$$

and then,

$$\frac{\lambda^\beta}{\lambda^\alpha} \geq \frac{W^\alpha_{b(a)} p_a}{W^\beta_{e(a)} p'_a} \tag{4}$$

Let us define

$$u(\alpha, \beta) = \max_{b(a) \in \mathcal{T}^\alpha, e(a) \in \mathcal{T}^\beta} \frac{W^\alpha_{b(a)} p_a}{W^\beta_{e(a)} p'_a}$$

If there is no arc from $\mathcal{T}^\alpha$ to $\mathcal{T}^\beta$, we set $u(\alpha, \beta) = 0$. We then build a valued reduced graph $R = (N, E)$ as follows:

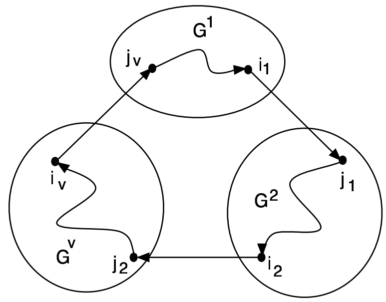

Figure 4: Definition of c

1. $N = \{0, \ldots, r\}$,
2. $\forall \alpha \in \{1, \ldots, r\}$, we build arc $(0, \alpha)$ with value $u(0, \alpha) = \lambda^{\alpha}_{\min}$,
3. A couple $(\alpha, \beta) \in \{1, ..., r\}^2$ belongs to E if and only if $u(\alpha, \beta) \neq 0$. The value of this arc is then $u(\alpha, \beta)$.

For every path μ of R, we set $U(\mu) = \prod_{e \in E \cap \mu}^{s} u(e)$. We prove then the following lemma:

Lemma 4.3. *For every circuit C of R, $U(C) < 1$.*

Proof. Let $C = 1, 2, ..., v, 1$ be a circuit of R. We can define the sequences i_s and j_s, $s = 1, ..., v$ of vertices of G such that :

1. i_s and j_{s-1} (*resp.* i_1 and j_v) are in the same unitary component G^s for $s \in \{2, ..., v\}$.
2. for $s \in \{1, ..., v\}$, there exist a sequences of arcs a_s with $b(a_s) = i_s, e(a_s) = j_s$ with $u(s, s+1) = \dfrac{W^s_{i_s} p_{a_s}}{W^{s+1}_{e(a_s)} p'_{a_s}}$ for $s < v$ and $u(v, 1) = \dfrac{W^v_{i_v} p_{a_v}}{W^1_{e(a_v)} p'_{a_v}}$.

So, we get a circuit c of G which does not belong to a unitary component of G, hence $\Pi(c) > 1$. Let us denote by ν_s, $s \in \{1, ..., u\}$ the sub-paths of c in the unitary component G^s. Then, $\Pi(c) = \prod_{s=1}^{u} \Pi(\nu_s) \prod_{s=1}^{u} \pi_{a_s}$. But by the definition of W and since G^s is a unitary graph, $\Pi(\nu_1) = \dfrac{W^1_{j_v}}{W^1_{i_1}}$ and

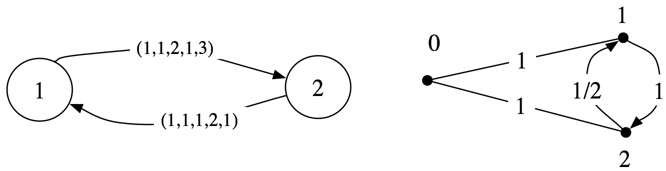

Figure 5: A linear precedence graph G and the corresponding reduced graph R

$\Pi(\nu_s) = \dfrac{W^s_{j_{s-1}}}{W^1_{i_s}}$ for $s > 1$. Moreover, by definition of u, $\pi_{a_s} = \dfrac{p'_{a_s}}{p_{a_s}}$, so we get $\pi_{a_s} = \dfrac{W^s_{i_s}}{W^{s+1}_{j_s} u(a_s)}$ for $s < v$ and $\pi_{a_u} = \dfrac{W^v_{i_v}}{W^1_{j_v} u(a_v)}$. So,

$$\Pi(c) = \prod_{s=1}^{v} \frac{1}{u(a_s)} = \frac{1}{U(C)} > 1$$

□

Theorem 4.4. *In any periodic schedule, the associated $\lambda^1, \ldots, \lambda^r$ meets the following system of inequalities:*

$$\forall (\alpha, \beta) \in \{1, \ldots, r\}^2, \quad \frac{\lambda^\beta}{\lambda^\alpha} \geq u(\alpha, \beta) \tag{5}$$

$$\forall \alpha \in \{1, \ldots, r\}, \quad \lambda^\alpha \geq \lambda^\alpha_{\mathrm{min}} \tag{6}$$

Moreover, there exists a minimum solution $\lambda^1_{\mathrm{opt}}, ..., \lambda^r_{\mathrm{opt}}$, i.e. *such that any other solution $\lambda^1, \ldots, \lambda^r$ satisfies:*

$$\forall \alpha \in \{1, \ldots, r\}, \quad \lambda^\alpha \geq \lambda^\alpha_{\mathrm{opt}}$$

Proof. The first part of the theorem is a simple outcome of equation 4 and lemma 4.2. From lemma 4.3, we can define, for all $\alpha \in \{1, \ldots, r\}$, $\lambda^\alpha_{\mathrm{opt}}$ to be the path with maximum U-value from node 0 to node α in the graph R. □

Hence, as we aim to minimize the periods, one can hope that there will exist a periodic schedule of G that is based on the vector $\lambda^1_{\mathrm{opt}}, ..., \lambda^r_{\mathrm{opt}}$. Unfortunately, this is not true in general. Indeed, let us consider the graph G with $\mathcal{T} = \{1, 2\}$ pictured by Figure 5. The two unitary components of G are $G^1 = (\{1\}, \emptyset)$ and $G^2 = (\{2\}, \emptyset)$. We get $W_1^1 = W_2^2 = 1$ and $\lambda^1_{\min} = \lambda^2_{\min} = 1$. From the reduced graph, we obtain then $\lambda^1_{\text{opt}} = \lambda^2_{\text{opt}} = 1$. On another hand, the linear programming system associated with G is:

$$\left\{\begin{array}{l} Min\ B \\ w_1 - 2w_2 \leq 0 \\ w_2 - w_1 \leq 0 \\ t_2 - t_1 \geq 1 - 2w_2 \\ t_1 - t_2 \geq 1 + w_2 \\ 1 \leq w_1 \leq B,\ 1 \leq w_2 \leq B \\ t_1 \geq 0,\ t_2 \geq 0 \end{array}\right. \tag{7}$$

and an optimal solution is given by $w_1 = w_2 = 2$, $t_1 = 3$ and $t_2 = 0$. So, there is no periodic schedule with period $\lambda^1_{\text{opt}} W_1^1 = \lambda^2_{\text{opt}} W_2^2 = 1$. However, in the following, we prove that, if a schedule exists and if the unitary components of the graph can be scheduled periodically, it is always possible to build a quasi-periodic schedule of G which starts with the earliest schedule, and becomes periodic according to the minimal vector $\lambda^1_{\text{opt}}, ..., \lambda^r_{\text{opt}}$ in the steady state.

Definition 4.5. A schedule σ is said to be quasi-periodic if

$$\forall i \in \mathcal{T}, \exists n_i^0 \geq 0 / \forall n \geq n_i^0, t^\sigma(< i, n >) = t_i + (n-1)w_i$$

Let us assume that the period of every task $i \in \mathcal{T}^\alpha$ is $\lambda^\alpha_{\text{opt}} W_i^\alpha$. If a is an arc of $A - \cup_{\alpha=1}^r A^\alpha$ with $b(a) \in \mathcal{T}^\alpha$ and $e(a) \in \mathcal{T}^\beta$, we obtain from lemma 2.2 the inequality:

$$\forall k \geq 0, t_{e(a)} - t_{b(a)} \geq l_a + \lambda^\alpha_{\text{opt}} W^\alpha_{b(a)}(p_a k + q_a - 1) - \lambda^\beta_{\text{opt}} W^\beta_j (p'_a k + q'_a - 1)$$

Setting $m_a = l_a + \lambda^\alpha_{\text{opt}} W^\alpha_{b(a)}(q_a - 1) - \lambda^\beta_{\text{opt}} W^\beta_{e(a)}(q'_a - 1)$ and $h_a = \lambda^\beta_{\text{opt}} W^\beta_{e(a)} p'_a - \lambda^\alpha_{\text{opt}} W^\alpha_i p_a$, we get

$$t_{e(a)} - t_{b(a)} \geq m_a - k h_a \tag{8}$$

Notice that for any arc a, $h_a \geq 0$.

We denote by $\mathcal{C}$ the set of circuits of G which weight is strictly greater than 1.

$$\mathcal{C} = \{\text{circuit} \quad \text{c} \quad \text{of} \quad \text{G}, \Pi(\text{c}) > 1\}$$

For every path ν of G, we also denote by $M(\nu) = \sum_{e \in \nu} m_e$ and by $H(\nu) = \sum_{e \in \nu} h_e$.

Lemma 4.6. *For every $c \in \mathcal{C}$, $\quad H(c) > 0$.*

Proof. Let $c \in \mathcal{C}$. Since $\Pi(c) > 1$, we can define the sequences i_s and j_s, $s = 1...v$ of vertices in the same way as in the proof of lemma 4.3. Then, we get

$$H(c) = \sum_{s=1}^{v} H(\nu_s) + \sum_{s=1}^{v} h_{a_s}$$

- We prove that $H(\nu_s) = 0$. Indeed, if ν_s is the sequence of arcs $x_1, ..., x_w$, we get

$$H(\nu_s) = \sum_{v=1}^{w} \lambda_{\text{opt}}^s (W_{e(x_v)}^s p'_{x_v} - W_{b(x_v)}^s p_{x_v})$$

  Now, by lemma 3.2, $\dfrac{W_{e(x_v)}^s}{W_{b(x_v)}^s} = \dfrac{p_{x_v}}{p'_{x_v}}$, so $H(\nu_s) = 0$.

- By definition of λ_{opt}^s, we get :

$$\forall s \in \{1, \ldots, v-1\} \frac{\lambda_{\text{opt}}^{s+1}}{\lambda_{\text{opt}}^s} \geq \frac{W_{i_s}^s p_{a_s}}{W_{j_s}^{s+1} p'_{a_s}} \quad \text{and} \quad \frac{\lambda_{\text{opt}}^1}{\lambda_{\text{opt}}^v} \geq \frac{W_{i_v}^s p_{a_v}}{W_{j_v}^1 p'_{a_v}} \tag{9}$$

  So, we obtain

$$\frac{\lambda_{\text{opt}}^1 ... \lambda_{\text{opt}}^v}{\lambda_{\text{opt}}^v \lambda_{\text{opt}}^1 ... \lambda_{\text{opt}}^{v-1}} = 1 \geq (\prod_{s=1}^{v} \frac{1}{\pi_{a_s}}) \frac{W_{i_1}^1 W_{i_2}^2 ... W_{i_v}^v}{W_{j_v}^1 W_{j_1}^2 ... W_{j_{v-1}}^v}$$

  By lemma 3.1 and by theorem 3.4, we know that $\Pi(\nu_s) = \dfrac{W_{j_{s-1}}^s}{W_{i_s}^s}$, $s = 2...v$ and $\Pi(\nu_1) = \dfrac{W_{j_v}^1}{W_{i_1}^1}$. So, the previous inequality can be rewritten:

$$1 \geq \prod_{s=1}^{v} \frac{1}{\pi_{a_s}} \prod_{s=1}^{v} \frac{1}{\Pi(\nu_s)} = \frac{1}{\Pi(c)}$$

  Now, we know that $\Pi(c) > 1$, so there is at least one arc (a_s) for which the inequality 9 is strict, so that $h(a_s) > 0$. Since all the others are nonnegative, we get the lemma. $\square$

By lemma 4.6, we can define k_0 as

$$k_0 = \max_{c \in \mathcal{C}} \left\lceil \frac{M(c)}{H(c)} \right\rceil$$

We set, for every $i \in \mathcal{T}$,

$$n_i^1 = \max(\max_{a=(i,j)\in A} p_a k_0 + q_a, \max_{a=(j,i)\in A} p'_a k_0 + q_a)$$

The idea is to build a quasi-periodic schedule σ as follows:

1. We start to execute the tasks following the earliest schedule. We stop at time t when, for every task i, the index of its last execution n is greater than or equal to n_i^1,$i.e$ when at least n_i^1 occurrences of i have been scheduled. Let us denote by $n_i^0, i \in V$ the index of the next execution of task i. We also denote by

   $$\mathcal{T}(k_0) = \{< i, n >, i \in \mathcal{T}, n < n_i^0\}$$

   the executions performed in this first phase.

2. In the second phase, the tasks are executed periodically. Let us denote by $\mathcal{T}^*$ the infinite set of the executions of tasks from $\mathcal{T}$.The infinite set of executions of tasks performed in this second phase is $\mathcal{T}^* - \mathcal{T}(k_0)$. We define $t_i \geq 0, i \in \mathcal{T}$ that meet the following requirements:

   - For any arc $a \in A$, $t_{e(a)} - t_{b(a)} \geq m_a - k_0 h_a$.
   - For every arc $a \in A$, and every couple of executions $< b(a), n >\in \mathcal{T}(k_0)$ and $< e(a), n' >\in \mathcal{T}^* - \mathcal{T}(k_0)$ with $e(a) \in \mathcal{T}^\beta$,

   $$t_{e(a)} \geq t^\sigma(< b(a), n >) + l_a - (n' - 1)\lambda_{\text{opt}}^\beta W_{e(a)}^\beta$$

   Notice that it follows from the definition of k_0 that such $t_i \geq 0, i \in \mathcal{T}$ exists and can be computed in polynomial time using Bellman-Ford's algorithm. For every task $i \in \mathcal{T}^\alpha$ we set

   $$\forall n \geq n_i^0, \quad t^\sigma(< i, n >) = t_i + (n - 1)\lambda_{\text{opt}}^\alpha W_i^\alpha$$

We get the following lemma by construction of the schedule:

Lemma 4.7. *For any arc $a \in A$, let us consider the executions $< b(a), n >$ and $< e(a), n' >$ such that $n = p_a k + q_a$ and $n' = p'_a k + q'_a$. If $< b(a), n >\in \mathcal{T}^* - \mathcal{T}(k_0)$ then $< e(a), n' >\in \mathcal{T}^* - \mathcal{T}(k_0)$.*

Proof. Since $< b(a), n >\in \mathcal{T}^* - \mathcal{T}(k_0)$, then all its successors according to the precedence constraints cannot have been scheduled by the earliest schedule. So $< e(a), n' >\in \mathcal{T}^* - \mathcal{T}(k_0)$. □

Theorem 4.8. *If the unitary components of a linear precedence graph G can be scheduled periodically, then there exists a quasi-periodic schedule such that every task $i \in \mathcal{T}^\alpha$, $\alpha \in \{1, ..., r\}$, has a minimum period $\lambda^\alpha_{\mathrm{opt}} W^\alpha_i$.*

Proof. We must check that the schedule built previously meets all the precedence constraints between the executions of the tasks. Let us consider an arc $a \in A$ with $b(a) \in \mathcal{T}^\alpha$, $e(a) \in \mathcal{T}^\beta$ and the executions $< b(a), n >$ and $< e(a), n' >$ with $n = p_a k + q_a$ and $n' = p'_a k + q'_a$. We consider 3 cases :

1. If $< b(a), n >\in \mathcal{T}^* - \mathcal{T}(k_0)$ then by lemma 4.7, $< e(a), n' >\in \mathcal{T}^* - \mathcal{T}(k_0)$. So,

   $$t^\sigma(< e(a), n >) = t_{b(a)} + (p_a k + q_a - 1)\lambda^\alpha_{\mathrm{opt}} W^\alpha_{b(a)}$$

   and

   $$t^\sigma(< e(a), n' >) = t_{e(a)} + (p'_a k + q'_a - 1)\lambda^\beta_{\mathrm{opt}} W^\beta_{e(a)}$$

   We prove that $t^\sigma(< e(a), n' >) \geq t^\sigma(< b(a), n >) + l_a$. Indeed,

   $$t^\sigma(< e(a), n >) - t^\sigma(< b(a), n' >) - l_a = t_{e(a)} - t_{b(a)} + kh_a - m_a$$

   Since $t_{e(a)} - t_{b(a)} \geq m_a - k_0 h_a$, we get

   $$t^\sigma(< e(a), n >) - t^\sigma(< b(a), n' >) - l_a \geq (k - k_0)h_a$$

   We know that $h_a \geq 0$. Moreover, since $< b(a), n >\in \mathcal{T}^* - \mathcal{T}(k_0)$ and $< e(a), n' >\in \mathcal{T}^* - \mathcal{T}(k_0)$ we get $k \geq k_0$, so the precedence constraint is met.

2. If $< b(a), n >\in \mathcal{T}(k_0)$ and $< e(a), n' >\in \mathcal{T}(k_0)$, then, they are both executed following the earliest schedule, according to their precedence constraint.

3. Lastly, if $< b(a), n >\in \mathcal{T}(k_0)$ and $< e(a), n' >\in \mathcal{T}^* - \mathcal{T}(k_0)$, by the definition of $t_{e(a)}$,

   $$t^\sigma(< e(a), n' >) = t_{e(a)} + (n' - 1)\lambda^\beta_{\mathrm{opt}} W^\beta_{e(a)} \geq t^\sigma(< b(a), n >) + l_a$$

□

Notice that the the construction of the reduced graph R and the computation of an optimum period is polynomial. But, we have no complexity results concerning the computation of an optimal quasi-periodic schedule: indeed, the number of tasks that must be scheduled according to the earliest schedule might not be bounded by a polynomial function.

5 Conclusion

We have proposed a polynomial algorithm to compute the existence of a periodic schedule on a linear precedence graph. If the graph is unitary, the computation turns out to be more simple and powerful. We have shown that in some cases, there might exist a schedule but not a periodic schedule.

The computation of the best periodic schedule appears as easy as the uniform case. However, the performance of this schedule with respect to the earliest schedule is still to be studied.

We have shown that if the graph is not unitary, then a better performance can be achieved by using quasi-periodic schedules, in which tasks becomes periodic after some executions. We have provided an algorithm for building optimal quasi-periodic schedules based on the decomposition of the linear graph into unitary components.

These results should be further generalized to handle schedules on generalized timed event-graphs and dataflow models.

However, the existence of a polynomial time feasibility test for a unitary graph remains open. It would be interesting to further analyse the complexity of this problem which we conjecture to be co-NP-hard.

In the next future, introduction of resource constraints should help to solve problems that occur in embedded systems.

References

[1] Jacques Carlier and Philippe Chrétienne. Timed petri nets schedules. In *Advances in Petri nets*, number 340 in Lectures Notes in Computer Sciences, pages 62–84. Springer Verlag, 1988.

[2] Guillaume Cavory, Remy Dupas, and Gilles Goncalves. A genetic approach to solving the problem of cyclic job shop scheduling with linear constraints. To appear in European Journal of Operational Research, 2003.

[3] Philippe Chrétienne. Transient and limiting behavior of timed event graphs. *RAIRO Techniques et Sciences Informatiques*, 4:127–192, 1985.

[4] Liliana Cucu and Yves Sorel. Schedulability condition for systems with precedence and periodicity constraints without preemption. In *RTS2003 11th conference on Real-time and embedded systems*, 2003.

[5] R. Govindarajan and G. Gao. Rate-optimal schedule for multi-rate dsp computations, 1993.

[6] Nicholas G. Hall, Tae-Eog Lee, and Marc Posner. The complexity of cyclic scheduling problems. *Journal of Scheduling*, 5:307–327, 2002.

[7] Claire Hanen and Alix Munier. Cyclic scheduling on parallel processors: An overview. In Philippe Chrétienne, Edward G. Coffman, Jan Karel Lenstra, and Zhen Liu, editors, *Scheduling theory and its applications*. J. Wiley and sons, 1994.

[8] Jan Korst, Emile Aarts, and Jan Karel Lenstra. Scheduling periodic tasks. *INFORMS journal on Computing*, 8(4):428–435, 1996.

[9] Edward Ashford Lee and David G. Messerschmitt. Synchronous dataflow. In *Procs. of the IEEE*, volume 75, pages 235–245, 1987.

[10] Tom Mc Cormick and Uday S. Rao. Some complexity results in cyclic scheduling. *Mathematical and Computer Modelling*, 20:107–122, 1994.

[11] Martin Middendorf and Vadim G. Timkovsky. On scheduling cycle shops: Classification, complexity and approximation. *Journal of Scheduling*, 5:135–169, 2002.

[12] Alix Munier. Régime asymptotique optimal d'un graphe d'événements temporisé généralisé: application à un problème d'assemblage. *RAIRO-Automatique Productique Informatique Industrielle*, 27(5):487–513, 1993.

[13] Alix Munier. The cyclic scheduling problem with linear precedence constraints. *Discrete Applied Mathematics*, 64:219–238, 1996.

[14] Paolo Serafini and Walter Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, pages 550–558, 1989.