Cyclic scheduling: an introduction

Claire Hanen
Claire.Hanen@lip6.fr

Université Paris X Nanterre / LIP6 UPMC
Summary

- Definition
  - Uniform precedence constraints
    - Feasibility
    - Periodic schedules
    - Earliest schedule
  - Resource constraints
    - Periodicity and Patterns
    - Complexity
    - approximation
    - ILP formulations
- Perspectives
Definition

A set of tasks $\mathcal{T}$ to be repeated many times assumed infinite

For $i \in \mathcal{T}$, $< i, k >$ denotes $k^{th}$ occurrence of $i$

An infinite schedule $\sigma$ defines:

$\forall k \geq 0 \ t^\sigma(<i,k>)$ starting time of $<i,k>$

Resources for each task execution
Optimizing

- Minimizing the average cycle time:

\[ A(\sigma) = \max_{i \in T} A(\sigma, i) = \limsup_{k \to +\infty} \frac{t^\sigma(<i,k>)}{k} \]

- Maximizing the throughput:

\[ D(\sigma) = \frac{1}{A(\sigma)} \]

- For a given average cycle time, minimizing the amount of resources per time units (ex: nb of processors, nb of registers,...)
Applications and models

- Implementing loops on parallel architectures
  → compiling, code generation
  → embedded applications
  literature on software pipelining and dataflow computations

- Mass production
  → Cyclic shop problems
  → Hoist scheduling problem.
  literature on cyclic scheduling in production systems

- Models of parallelism
  → timed Petri Nets
  → Graphs
  → Max + algebra
  literature on parametric paths, timed event graphs, Max +
Infinite schedule?

- Algorithms -> finite description
- Static schedule
  - Must be regular /time and resources
- Dynamic policy
  - examples: earliest schedule, regular priorities on resources assignment
Loop example

Assume arrays $A, B, C, D$, and that the processor can compute several instructions in parallel:

for $I = 2$ to $N$

\begin{align*}
B(I) &= A(I - 1) + 1 & \text{task 1} & & < 4, k - 1 > \text{ precedes } < 1, k > \\
C(I) &= B(I) + 5 & \text{task 2} & & < 1, k > \text{ precedes } < 2, k > \\
D(I) &= B(I - 2) \ast D(I) & \text{task 3} & & < 1, k - 2 > \text{ precedes } < 3, k > \\
A(I) &= C(I - 2) + D(I) & \text{task 4} & & < 2, k - 2 >, < 1, k > \text{ precede } < 2, k >
\end{align*}

Precedence constraints are uniforms: if $< i, k > \text{ precedes } < j, l >$ then for all integers $\delta$, $< i, k + \delta > \text{ precedes } < j, l + \delta >$
Cyclic scheduling with uniform precedences

- A set $\mathcal{T}$ of $n$ generic tasks with processing times $p_1, \ldots, p_n$
- non reentrance: $<i, k>$ precedes $<i, k+1> \ \forall k \geq 1$
- A multi-graph $G = (\mathcal{T}, A)$ of uniform constraints
- For each arc $a \in A$,
  - A value called length $L(a) \in \mathbb{Z}$ also mentioned as latency or delay
  - A value called height $H(a) \in \mathbb{Z}$ also called dependence distance

\[
(L(a), H(a)) \quad \text{If } i = b(a) \text{ et } j = e(a), \forall k \geq 1, t^\sigma(<i, k>) + L(a) \leq t^\sigma(<j, k + H(a)>)
\]

Find an infinite feasible schedule minimizing the average cycle time.

**Definition 1.** The Length of path $\mu$, denoted $L(\mu) =$ sum of arcs length. Height of $\mu$, denoted by $H(\mu) =$ sum of arcs height.

**Remark 1.** Non-reentrance can be expressed with uniform constraints : arcs $(i, i)$ with length $p_i$ and height 1.
Negative length can be used to model deadlines:

\[ t^\sigma(\langle 5, k \rangle) \leq t^\sigma(\langle 4, k \rangle) + 10 \]
Questions

- Feasibility
- Construction and properties of schedules.
  Periodic schedules
  Earliest schedule
- Performance
Feasibility

For any path \( \mu \) from \( i \) to \( j \),

\[
\forall k \geq \max(1, 1 - H(C)), \quad t^\sigma(< i, k >) + L(\mu) \leq t^\sigma(< j, k + H(\mu) >)
\]

For any circuit \( C \) of \( G \), and task \( i \) in \( C \):

\[
\forall k \geq \max(1, 1 - H(C)), \quad t^\sigma(< i, k >) + L(C) \leq t^\sigma(< i, k + H(C) >)
\]

Non reentrance \( \Rightarrow \)

**Lemma 1.** [Lee 05][Munier 06] If \( G \) is feasible then for any circuit \( C \) such that \( H(C) \leq 0 \), \( L(C) \leq 0 \). If \( H \in \mathbb{N} \) this condition is sufficient [many authors] [Chretienne 85]

- If \( H(C) > 0 \), \( k = qH(C) + r \) then \( t^\sigma(< i, k >) \geq qL(C) + t^\sigma(< i, r >) \).
- If \( H(C) < 0 \) and \( k = -qH(C) + r \) then \( t^\sigma(< i, k >) \leq -qL(C) + t^\sigma(< i, r >) \)

**Lemma 2.** If \( H(C) \geq 0 \), \( A(\sigma, i) = \limsup_{k \to +\infty} \frac{t^\sigma(< i, k >)}{k} \geq \frac{L(C)}{H(C)} \).

If \( H(C) < 0 \), \( A(\sigma, i) \leq \frac{L(C)}{H(C)} \).
For a circuit $C$ the Index of $C$: $\alpha(C) = \frac{L(C)}{H(C)}$.

- $C^+(i)$ set of circuits $C$ s.t. $i \in C$ and $H(C) \geq 0$. $C^+ = \bigcup_{i \in T} C^+(i)$
- $C^-(i)$ set of circuits $C$ s.t. $i \in C$ and $H(C) < 0$. $C^- = \bigcup_{i \in T} C^-(i)$

**Corollary 1.** If $G$ is feasible then for any task $i$, and for any schedule $\sigma$

\[
\max_{C \in C^+(i)} \alpha(C) \leq A(\sigma, i) \leq \min_{C \in C^-(i)} \alpha(C)
\]

A critical circuit is a circuit $C^* \in C^+$ s.t. $\alpha(C')$ is maximum: lower bound on $A(\sigma)$.
Definition 2. A schedule $\sigma$ is periodic if each task $i$ has a period $w_i$ such that:

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

Let $a$ be an arc from $b(a)$ to $e(a)$

$$\forall k, \quad t^\sigma(< b(a), 1 >) + (k - 1)w_{b(a)} + L(a) \leq t^\sigma(< e(a), 1 >) + (k - 1)w_{e(a)} + H(a)w_{e(a)}$$

$$\Leftrightarrow t^\sigma(< e(a), 1 >) - t^\sigma(< b(a), 1 >) \geq L(a) - w_{e(a)}H(a) + (k - 1)(w_{b(a)} - w_{e(a)})$$
Lemma 3. for any arc $a$, $w_{b(a)} \leq w_{e(a)}$.

We denote by $C_1, \ldots, C_q$ the strong components of $G$.

Let $\alpha^+(C_s) = \max_{C \in C_s, H(C) > 0} \alpha(C)$ and $\alpha^-(C_s) = \min_{C \in C_s, H(C) < 0} \alpha(C)$

Corollary 2.

$$\forall s, \forall i, j \in C_s, w_i = w_j = W_s \quad \text{and} \quad \alpha^+(C_s) \leq W_s \leq \alpha^-(C_s)$$

Moreover, if there is an arc $a$ s.t. $b(a) \in C_s, e(a) \in C_s'$ then $W_s \leq W_{s'}$
Example

\[ C_1 = \{1, 2, 3\}, C_2 = \{4, 5\}, C_3 = \{6\} \]

\[ \alpha^+(C_1) = 10, \alpha^-(C_1) = 17, \quad \alpha^+(C_2) = 7, \alpha^-(C_2) = 11, \quad \alpha^+(C_3) = 5 \]

\[ 10 \leq W_1 \leq 17, \quad 7 \leq W_2 \leq 11, \quad 5 \leq W_3, \]

\[ W_3 \geq W_2, W_3 \geq W_1 \]

\[ W_1 = 10, W_2 = 7, W_3 = 10. \text{ Notice that } W_1 = W_2 = W_3 = 10 \text{ is also a solution.} \]
Feasibility

Theorem 1. [Munier 06] $G$ is feasible if and only if there exists a periodic schedule.

Idea: If no periodic schedule exists, build paths $\mu_x$ between $i$ and $j$ in $G$ such that $H(\mu_x) = h$ and $\lim_{x \to +\infty} L(\mu_x) \to +\infty$, which contradicts

$$t^\sigma (\langle j, k \rangle) - t^\sigma (\langle i, k + h \rangle) \geq L(\mu_x)$$

for some $k$. 
Computation of a periodic schedule

For a strong connected graph $G$ and a given $W$:

$$\forall a, \quad t^\sigma(< e(a), 1 >) - t^\sigma(< b(a), 1 >) \geq L(a) - W.H(a)$$

- Let $V_W(a) = L(a) - W.H(a)$
- If $(G, V_W)$ has positive circuits then infeasibility.
- otherwise $t^\sigma(< i, 1 >) =$ longest path to $i$ in $(G, V_W)$ is a solution.
- Bellman-Ford algorithm
Computation of critical circuits

[Dasdan et al 99]

- Polynomial algorithms:
  - Binary search on $W$: at each step check if $(G, V_W)$ has positive circuits $O(nm(\log n + \log \max_a (L(a), H(a))))$ [Lawler 79][Gondran-Minoux 85]
  - Linear programming with primal dual approach $O(n^2 m)$ [Burns 91]

- An efficient pseudo polynomial algorithm: Howard’s algorithm [Cochet-Terrasson et al 98]

  $W =$ lower bound
  - At each step check if $(G, V_W)$ has positive circuit $C$ with breadth first search. If $H(C) \leq 0$ stop. Otherwise set $W = \alpha(C)$.

  complexity $O(m.X)$, $X$ product of degrees of nodes.
Computation of an optimal periodic schedule

- Compute the strong components of $G$. $O(n + m)$
- Check feasibility for each component $C_s$, and the critical circuit value $\alpha_s$. $O(nm \log(n) + \log(V_{max}))$
- Compute the reduced graph of components. $O(n + m)$
- Sort the components by topological order. $O(n + m)$
- for each component $C_s$.
  - if $C_s$ has no predecessor, set $W_s = \alpha_s$.
  - if $C_{x_1}, \ldots, C_{x_r}$ are predecessors of $C_s$. Let $\beta_s = \max_i W_{x_i}$
  - If $\alpha_s \geq \beta_s$ set $W_s = \alpha_s$.
  - otherwise check if $(C_s, V_{\beta_s})$ has some positive circuit. $O(nm)$
  - if so, infeasibility otherwise set $W_s = \beta_s$.
- Compute the longest paths from a dummy source node to any node $i$ on $G$ with on each component value $V_{W_s}$ and on each intermediate arc between $C_s, C_s'$ value $V_{W_{s'}}$. $O(nm)$
- $t^\sigma(\langle i, 1 \rangle) =$ longest path to $i$. 
Example

Graph with $L - W_sH$ values:

A periodic schedule: period of 1, 2, 3, 6 is 10, period of 4, 5 is 7
Positive heights

The problem has been studied earlier [Chretienne 85][many authors]

**Theorem 2.** For any $w \geq \max_{C \text{ circuit}} \alpha(C)$ there is a periodic schedule, all tasks have period $w$.

No need to compute strong components.
**K-periodic schedules**

**Definition 3.** In a $K$-periodic schedule $\sigma$, a schedule of $K_i$ occurrences of task $i$ is repeated every $W_i$ time units:

$$\forall l \geq l_0, \quad t^\sigma(<i, l + K_i>) = t^\sigma(<i, l>) + W_i$$

**Lemma 4.** The average cycle time of task $i$ in a $K$—periodic schedule is $A(\sigma, i) = \frac{W_i}{K_i}$.

Un ordonnancement 2-périodique (pour chaque tâche).

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>4</th>
<th>1</th>
<th>7</th>
<th>3</th>
<th>4</th>
<th>1</th>
<th>3</th>
<th>4</th>
<th>1</th>
<th>7</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>
Properties of the earliest schedule

[Romanovskii 67, Chretienne 85, Munier 06]

**Theorem 3.** If $G$ is feasible, the earliest schedule $\sigma^*$ is $K$-periodic and

$$A(\sigma^*) = \max_{C \in \mathcal{C}^+} \alpha(C')$$

After a transitory time, the earliest schedule becomes regular and its behavior is ruled by the critical circuit index.

**Remark 2.** $K$ might be large $\leq$ product of heights of critical circuits.

**Open questions**

- Length of the transitory time?
- Exact measure of $K$. 
Stability/boundedness

If $w < w'$ then $b(a)$ produces pieces or results faster than $e(a)$.

Unstability
Moreover the iteration delay:

$$D^\sigma(k) = \max_{i \in T} t^\sigma(<i, k>) - \min_{i \in T} t^\sigma(<i, k>) \to k \to +\infty + \infty$$

This might occur in earliest schedule and in periodic schedules for general uniform graphs.
If task 4 has processing time 1:

- There are 3 strong components: \( C_1 = \{1, 2, 3\}, C_2 = \{4\}, C_3 = \{5, 6, 7\} \)
- \( \alpha(C_1) = 3, \alpha(C_2) = 1, \alpha(C_3) = 5 \)
- The earliest schedule of \( i \leq 4 \) is 2-periodic with period 6.
- The earliest schedule of \( \{5, 6, 7\} \) is 1-periodic with period 5.
- Unstable schedule
- Here a static 1-periodic schedule with period 5 can be built.
Cyclic problems with resources

- Complexity of a cyclic problem/ its non cyclic version.
- Tools for constructing periodic schedules: circuits and patterns
  Polynomial problems
- Are periodic schedules optimal?
- Decomposed scheduling: a general approach
- Approximation of decomposed scheduling
- ILP formulations
Complexity

Consider a problem $resources|prec|C_{max}$ which is NP-hard.

A simple Reduction:

- Add two dummy nodes to $G$ (source $s$, sink $t$ with unit processing set $H(a) = 0$ for all arcs in $G$ and arcs from $s$ and arcs to $t$.
- Add an arc from $t$ to $s$ with height 1.
- Any schedule $\sigma'$ of $G'$ is a sequence of schedules of $G$.

Its average cycle time $A(\sigma')$ is the mean of makespan of $G$ schedules

$A(\sigma') \leq B \iff C_{max}(G) \leq B - 2$

**Corollary 3.** $P|uniform~prec, p_i = 1|A, pre - assigned ~processors|uniform ~prec|A$, cyclic job-shop with uniform constraints are NP-hard
Circuits

Consider a uniform graph $G = \text{circuit}$, such that for any arc $a$ $L(a) = p_{b(a)}$: usual precedence constraints.

**Lemma 5.** In any schedule of $G$ no more than $H(G)$ tasks are performed in parallel.

**Corollary 4.** [Munier 91] The problem $P|\text{circuit}, L(a) = p_{b(a)}|A$ is solvable in polynomial time.

Idea: if $H(G) \leq m$ then any schedule meets the resource constraint. If $H(G) > m$, then we can reduce the height of some arcs so that $H(G) = m$ without modifying the lower bound on

$$A(\sigma) \geq \max(\max_{i \in T} p_i, \frac{\sum_{i \in T} p_i}{m})$$
Let $\sigma$ be a periodic schedule with unique period $w$.

\[ t^\sigma(<i, k>) = t^\sigma(<i, 1>) + (k - 1)w \]

**Definition 4.** The **Pattern** of $\sigma$ is defined by: $\pi^\sigma(i) = t^\sigma(<i, 1>) \mod w$. The **iteration setting** of $\sigma$ is $\eta^\sigma(i) \in \mathbb{Z}$, s.t.

\[ t^\sigma(<i, 1>) = \pi^\sigma(i) + \eta^\sigma(i)w \]
Patterns

The pattern defines the schedule of tasks in an interval 
$[kw, (k + 1)w]$ for enough large $k$

The iteration setting indicates which occurrences of tasks are involved in this interval

In the interval $[kw, (k + 1)w]$, $i$ starts at $kw + \pi^\sigma(i)$ its occurrence $< i, k + 1 \ - \ \eta^\sigma(i) >$
Feasibility of a pattern

Question: Given a pattern $\pi$, and a period $w$ is there an iteration setting $\eta$ such that the uniform constraints are met by the periodic schedule?

For an arc $a$, $t(< e(a), 1 >) - t(< b(a), 1 >) \geq L(a) - wH(a)$.

**Lemma 6.** An iteration setting satisfies for any arc $a$

$$\eta(e(a)) - \eta(b(a)) \geq \left[ \frac{L(a) + \pi(b(a)) - \pi(e(a))}{w} \right] - H(a) = E_{w,\pi}(a)$$

**Lemma 7.** An iteration setting exists iff $(G, E_{w,\pi})$ has no positive circuit. can be checked in polynomial time
Polynomial problems

Corollary 5. if $G$ has no other circuit than the non-reentrance loops, any pattern has an iteration setting.

Theorem 4. [Munier 91] $P|\text{acyclic uniform prec}|A$ is solvable in polynomial time.

Idea: Consider tasks as independent, and schedule them on $m$ processors using Mc-Naughton algorithm (preemptive scheduling). Use the schedule as a Pattern of a periodic schedule and compute the iteration setting.

Theorem 5. dedicated processors $|\text{acyclic uniform prec}|A$ is solvable in polynomial time. In particular non-reentrant job-shop or flow-shop.

Idea: schedule operations on each machine as independent. Set $W = C_{max}$ (which is a lower bound on $A(\sigma)$ in this case). Use the schedule as a Pattern of a periodic schedule and compute the iteration setting.

[Robert, Legrand] also used similar ideas to build schedules for a broadcasting problem on a heterogeneous platform with net contentions.
Example

Cyclic scheduling for EPIT 2007 – p. 33/?
Optimality of periodic schedules?

In general, periodic schedules are not optimal schedules. Example: cyclic problem with two processors

However, periodic schedules are simple to implement and easier to compute.
Decomposed scheduling: a general approach

Also known as Decomposed software pipelining [Eisenbeis, Darte, Gasperoni, Schwiegelsohn, de dinechin, Munier,...]

- Build a non cyclic schedule $S$ of $T$ that meets the resource constraints and eventually some precedence constraints.
- Consider this schedule as a pattern, and set $W = C_{\text{max}}(S)$
- Build a feasible iteration setting for the pattern. (if possible)

OR

- Build an iteration setting.
- Deduce from the iteration setting precedence relations between tasks in a pattern
- Build a non cyclic schedule $S$ of $T$ that meets the resource constraints and these precedence constraints.
- Consider this schedule as a pattern, and set $W = C_{\text{max}}(S)$
Gasperoni-Schwiegelsohn algorithm

An algorithm for $P|\text{uniform with } L(a) = p_{b(a)}|A$.

- Compute an optimal schedule $\sigma^\infty$ on infinitely many processors.

- Consider the pattern $\pi^{\sigma^\infty}$ and remove arcs $a$ from $G$ such that $\pi^{\sigma^\infty}(b(a)) + p_{b(a)} > \pi^{\sigma^\infty}(e(a)) \rightarrow G'$.

- Schedule $G'$ on the $m$ processors according to a list scheduling algorithm $\rightarrow$ schedule $S$.

- Set $S$ as a pattern with $w = C_{\max}(S)$ and combine with the iteration setting of $\sigma^\infty \rightarrow$ periodic feasible schedule $\sigma$. 
Example

$G'$ has 3 arcs

Schedule on 2 processors

periodic schedule with period 6
Bounds

**Theorem 6.** [Gasperoni-Schwiegelsohn]

\[ A(\sigma) \leq (2 - \frac{1}{m})A(\sigma^{opt}) + \max_{i \in T} p_i \]

[Darte et al] generalized the algorithm by choosing a convenient \( G' \) using retiming.
Disjunctive constraints

Assume that two tasks \( i \) and \( j \) use the same resource: \( \forall k, l, \) either
\[
\sigma(< i, k >) + p_i \leq \sigma(< j, l >) \quad \text{or} \quad \sigma(< j, l >) + p_j \leq \sigma(< i, k >).
\]
- If \( \sigma \) is periodic with period \( w \),
  - in time interval \([xw, (x + 1)w)\]
    - there is only one occurrence \(< i, u_i(x) >\).
  - \(< j, u_i(x) + h_{ij} >\) next occurrence of \( j \).
  - \(< j, u_i(x) + h_{ij} - 1 >\) precedes \(< i, u_i(x) >\).

Hence setting \( h_{ij} \in \mathbb{Z} \) as a variable, the disjunctive constraints can be expressed as:

\[
\begin{cases}
    \sigma(< j, 1 >) - \sigma(< i, 1 >) \geq p_i - \omega h_{ij} \\
    \sigma(< i, 1 >) - \sigma(< j, 1 >) \geq p_j - \omega h_{ji} \\
    h_{ij} + h_{ji} = 1
\end{cases}
\]
ILP model

General form of constraints:

\[ t^\sigma(\langle e(a), 1 \rangle) - t^\sigma(\langle b(a), 1 \rangle) \geq L(a) - wH(a) \]

Heights might be either fixed or variables. Additional linear constraints on variable heights might be added.

**Remark 3.** For a fixed \( w \), the constraint is linear.

**Remark 4.** Once disjunctive variables of this ILP are fixed, it remains a uniform graph scheduling problem.

Many problems with disjunctive resources (shop-problems, but also Hoist scheduling problems) ([Roundy], [Brucker], [Levner et al] [Chu et al] [Lei et al]) can be formulated, for a fixed \( w \) as an mixed integer linear program.
Algorithms for disjunctive problems

- Branch and bound algorithms
  For cyclic job-shop problems [Roundy, Hanen, Kampmeyer and Brücker]
  For hoist scheduling [Lei and Wang, Chu and Proth]
- Metaheuristics [Kampmeyer and Brücker]
Some Open problems

- Transitory state of the earliest schedule.
- Complexity of the cyclic problem with unit lengths and 2 processors.
- How to use results on scheduling problems in their cyclic version?
  - Efficiency of Jackson algorithm for the 1-machine problem
  - Specific list schedules for parallel machines problems.
- Defining regular dynamic policy and study their behaviour.
  - Regularity of the schedule, performance bounds.