Problèmes d'ordonnancement en informatique

Safia Kedad-Sidhoum, Claire Hanen, Christophe Picouleau,

Claire.Hanen@lip6.fr

UPMC (LIP6) / Université paris X Nanterre (LIP6) / CNAM (CEDRIC)

Ordonnancement et parallélisme

- il y a plusieurs unités de traitement
- une tâche est un traitement indivisible pouvant être effectué sur une, plusieurs ou toutes les unités de traitement
- la durée d'une tâche est fixe
- les tâches produisent des résultats de calculs qui sont stockés et utilisés par d'autres tâches.
- Cela donne lieu à des dépendances de données: dépendances directes, antidépendance, dépendance en assignation
- les dépendances peuvent prendre un temps variable (problèmes à délais de communication) et des ressources (problème d'allocation de registres).

Problèmes récents

- Problèmes où les tâches se répètent un très grand nombre de fois (C. Hanen)
  - parallélisation de boucle
  - production en série d'un objet manufacturé.
- Problèmes où les temps de communication dépendent de l'affectation des tâches aux processeurs (C. Picouleau)
- Problèmes de plannification (S. Kedad-Sidhoum)

Problèmes répétitifs/cycliques

- Un ensemble de tâches génériques $\mathcal{T}$ de durées $p_i, i \in \mathcal{T}$ devant être exécutées un très grand nombre de fois.
- $< i, k >$ désigne la k^{ieme} occurrence de la tâche $i \in \mathcal{T}$
- Un ordonnancement σ définit une date de début $t^\sigma(< i, k >)$ pour chaque occurence de chaque tâche i.
- On définit C_N^σ date de fin de l'itération N:

$$C_N^\sigma = \max_{k \leq N} t^\sigma(< i, k >) + p_i$$

- Objectif: Minimiser le Temps de cycle moyen:

$$\alpha(\sigma) = limsup_{N \to \infty} \frac{C_N^\sigma}{N}$$

- Débit d'un ordonnancement: $\frac{1}{\alpha(\sigma)}$
- on notera α^{opt} le temps de cycle min d'un ordonnancement.

Contraintes d'un problème répétitif

- Contraintes génériques: s'appliquent de manière identique (ou régulière) aux différentes occurrences d'une tâche
- précédences généralisées
  - Précédences uniformes
  - précédences linéaires
- Contraintes de ressources (ex: machines parallèles)

Problématiques

- Déterminer la complexité des problèmes
- Pour les problèmes sans contrainte de ressource
  - Etudier les propriétés de l'ordonnancement "au plus tôt"
  - Trouver un ordonnancement "générique" optimal simple à décrire.
- Pour les problèmes avec contraintes de ressource
  - Trouver des politiques dynamiques d'ordonnancement efficaces
  - Construire des ordonnancements de structure simple (périodique).
  - Analyser les performances des ordonnancements obtenus.

Parallélisation d'une boucle

Supposons A, B, C, D vecteurs stockés dans des tableaux, et que l'architecture peut exécuter plusieurs instructions en même temps (ex: il y a plusieurs unités de traitement) :

pour I de 2 à N faire

$B(I) = A(I-1) + 1$	tâche 1	$< 4, k-1 >$ précède $< 1, k >$
$C(I) = B(I) + 5$	tâche 2	$< 1, k >$ précède $< 2, k >$
$D(I) = B(I-2) * D(I)$	tâche 3	$< 1, k-2 >$ précède $< 3, k >$
$A(I) = C(I-2) + D(I)$	tâche 4	$< 2, k-2 >, < 1, k >$ précèdent $< 2, k >$

Les précédences sont uniformes: si $< i, k >$ précède $< j, l >$ alors pour tout entier δ,

$< i, k+\delta >$ précède $< j, l+\delta >$

Contraintes uniformes

- n tâches génériques de durées $p_1, \ldots, p_n$ fixées
- les tâches sont non réentrantes $< i, k >$ précède $< i, k+1 >$ pour tout $k \geq 1$
- Un multi-graphe $G = (\mathcal{T}, A)$ de contraintes uniformes
- Pour chaque arc $a \in A$, une hauteur $H(a) \in \mathbb{N}$
- On note, pour un arc $a \in A$, $b(a)$ son origine et $e(a)$ son extrémité.
- si l'arc a est entre $i = b(a)$ et $j = e(a)$, $\forall k \geq 1$, $< i, k >$ précède $< j, k + H(a) >$
- On cherche un ordonnancement infini qui respecte ces contraintes et minimise le temps de cycle moyen.

Définition 1. *On appelle longueur d'un chemin μ de G et l'on note $L(\mu)$ la somme des durées des tâches origine de ses arcs. On appelle hauteur de μ et l'on note $H(\mu)$ la somme des hauteurs de ses arcs.*

Observation 1. *On peut assimiler les contraintes de non-réentrance à des arcs de G de i à i de hauteur 1 pour toute tâche i. On supposera implicitement ces arcs dans la suite.*

Un autre modèle

Graphes d'évènements T-temporisés.

- transistions= tâches génériques
- durée de franchissement= durée de la tâche
- places= précédences uniformes
- marquage initial = hauteur

Graphe développé $\mathcal{G}$

- sommets: $\{< 0, 0 >\} \cup \{< i, k >, \quad k \geq 1, i \in \mathcal{T}$
- Pour tout $i \in \mathcal{T}$, arcs de $< 0, 0 >$ à $< i, 1 >$ valués 0
- pour toute tâche $i \in \mathcal{T}$, pour tout $k \geq 1$, arc de $< i, k >$ à $< i, k+1 >$ valué p_i
- pour tout arc a de G d'origine i et d'extrémité j, pour tout $k \geq 1$, arc de $< i, k >$ à $< j, k + H(a) >$ valué p_i
- valuation notée v.

Observation 2. *Tout ordonnancement σ définit un ensemble de potentiels sur le graphe développé $\mathcal{G}$*

Faisabilité

Theorème 1. *Un ensemble de potentiels sur* $\mathcal{G}$ *existe ssi* $\mathcal{G}$ *ne comporte pas de circuits de valeur strictement positive. Dans ce cas, si* $t^{\sigma^*}(i,k)$ *est le chemin de valeur maximale de* $< 0, 0 >$ *à* $< i, k >$*,* σ^* *définit un ensemble de potentiels sur* $\mathcal{G}$*.*

Theorème 2. *Il existe un ensemble de potentiels sur* $\mathcal{G}$ *si et seulement si le graphe* G *ne comporte pas de circuit de hauteur négative ou nulle.*

Theorème 3. *Les ordonnancements constituent un treillis. Ce treillis admet un plus petit élément qui est l'ordonnancement* σ^* *(aussi appelé ordonnancement au plus tôt).*

Circuits de G

Observation 3. *si μ est un chemin de G passant d'origine i, et d'extrémité j, alors dans tout ordonnancement σ:*

$$\forall k \geq 1, t^\sigma(< j, k + H(\mu) >) \geq t^\sigma(< i, k >) + L(\mu)$$

Corollaire 1. *si μ est un circuit de G passant par un sommet i, alors dans tout ordonnancement σ:*

$$\forall k \geq 1, t^\sigma(< i, k + H(\mu) >) \geq t^\sigma(< i, k >) + L(\mu)$$

Theorème 4. *pour tout circuit μ de G,*

$$\alpha^{opt} \geq \frac{L(\mu)}{H(\mu)}$$

Circuits critiques

Définition 2. *On note*

$$\alpha(G) = \max_{\nu \text{ } circuit \text{ } de \text{ } G} \frac{L(\nu)}{H(\nu)}$$

Un circuit μ est dit critique si $\frac{L(\mu)}{H(\mu)} = \alpha(G)$.

Theorème 5. *Si G est faisable, alors il existe un circuit critique dans G.*

De plus, tous les circuits élémentaires de sa décomposition sont critiques

Ordonnancement périodique

Définition 3. *Un ordonnancement σ est dit périodique (ou 1-périodique) lorsqu'il existe un vecteur $t_i^\sigma,\ i \in \mathcal{T}$ et un nombre w^σ tel que pour toute tâche i et tout entier k:*

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

Theorème 6. *Tout ordonnancement périodique σ satisfait le système d'équations:*

$$\begin{cases} \forall\, a \in A & t^\sigma_{e(a)} - t^\sigma_{b(a)} \geq L(a) - w^\sigma H(a) \\ \forall i \in \mathcal{T} & 0 \leq t_i^\sigma \end{cases} \tag{1}$$

Ordonnancement périodique optimal

Theorème 7. *Il existe un ordonnancement périodique de période* $\alpha(G)$*. De plus, pour tout* $w \geq \alpha(G)$ *on peut construire un ordonnancement périodique de période* w *en cherchant un potentiel sur* G *muni de la valuation* $L - wH$ *appelée amplitude relative à* w*.*

Corollaire 2. *On peut calculer un ordonnancement périodique optimal en* $O(n^3 log n)$

Analyse de l'ordonnancement au plus tôt

Observation 4. *Caractériser σ^* c'est caractériser les plus longs chemins sur le graphe développé $\mathcal{G}$ sans développer le graphe (en travaillant sur G)*

Notations:

- $\mathcal{C}(i,j)$: ensemble des chemins de i à j dans G
- $\mathcal{C}_m(i,j)$ ensemble des chemins de i à j dans G de hauteur m
- $\lambda_m(i,j)$ longueur maximale d'un chemin de $\mathcal{C}_m(i,j)$

Proposition 1. *Soit G' le graphe construit en ajoutant à G un sommet origine 0 et des arcs de ce sommet à tous de hauteur 1 et de longueur 0. A tout chemin de G' de hauteur m issu de 0 et d'extrémité i est associé un chemin de $\mathcal{G}$ de même longueur, d'origine $< 0, 0 >$ et d'extrémité $< i, m >$, et réciproquement.*

Corollaire 3.

$$\forall i_i n \mathcal{T}, \forall k \geq 1, \quad t^{\sigma^*}(< i, k >) = \lambda_k(0, i)$$

Calcul du régime limite

Soient i_0, j_0 des sommets de G. On suppose que le graphe G restreint aux chemins de $\mathcal{C}(i_0, j_0)$ comporte un unique circuit critique ρ^*. Soit k_0 un sommet quelconque de ρ^*. On note :

- μ_1 un chemin de i_0 à k_0
- μ_2 un chemin de k_0 à j_0
- $\lambda_1 = L(\mu_1), \lambda_2 = L(\mu_2), h_1 = H(\mu_1), h_2 = H(\mu_1)$
- On note ν_k le chemin construit en empruntant μ_1, puis k fois ρ^* puis μ_2
- $s_k = H(\nu_k)$.

Lemme 1. *Il existe un entier K_0 tel que $\forall k \geq K_0$, le chemin de longueur maximale de $\mathcal{C}_{s_k}(i_0, j_0)$ passe par ρ^*.*

Lemme 2. *Si pour une valeur de $k \geq K_0$, $\lambda_{s_k}(i_0, j_0) = L(\mu_1' \rho^* \mu_2')$ alors $\lambda_{s_{k+1}}(i_0, j_0) = L(\mu_1'(\rho^*)^2 \mu_2') = \lambda_{s_k}(i_0, j_0) + L(\rho^*)$.*

Suites périodiques

Soit $u_k, k \geq 1$ une suite de nombres réels

Définition 4. *La suite u_k est dite K-périodique s'il existe deux nombres : $K \in IN, W \in IR$ et un entier k_0 tels que:*

$$\forall k \geq k_0, \quad u_{k+K} = u_k + W$$

On dit que K est le facteur de périodicité de la suite , W est sa période et $\frac{K}{W}$ est sa fréquence (l'inverse s'appelle également le temps de cycle moyen)

Lemme 3. *Si u_k et v_k sont deux suites respectivement K-périodiques de période W et K'-périodique de période W' alors*

$$\frac{K'}{W'} \geq \frac{K}{W} \Rightarrow w_k = \max(u_k, v_k) \quad est\ K - périodique\ de\ période\ W$$

Périodicité des plus longs chemins

Lemme 4. *La suite* $u_k = \lambda_{s_k}(i_0, j_0)$ *est 1-périodique de période* $L(\rho^*)$

Lemme 5. *En posant* $s_k^r = s_k + r$ *pour* $0 \leq r < H(\rho^*)$*, la suite*
$u_k^r = \lambda_{s_k^r}(i_0, j_0), \quad k \geq 1$ *est 1-périodique de période* $L(\rho^*)$

Theorème 8. *La suite* $\lambda_k(i_0, j_0)$ *est* $H(\rho^*)$*-périodique de période* $L(\rho^*)$

Cas de plusieurs circuits critiques

Theorème 9. *La suite* $\lambda_k(i_0, j_0)$ *est* K*-périodique de période* W *avec :* K*=ppcm des hauteurs de tous les circuits critiques.*

$$\frac{W}{K} = \alpha(\mathcal{C}(i_0, j_0))$$

Voir "Problèmes d'ordonnancement", J. Carlier, P. Chretienne

G fortement connexe

Si G est fortement connexe alors le sous-graphe induit par les chemins de i_0 à j_0 est toujours G pour tout couple (i_0, j_0). On en déduit:

Theorème 10. *L'ordonnancement au plus tôt σ^* est K-périodique de période W avec $\frac{W}{K} = \alpha(G)$. Toutes les suites $t^{\sigma^*}(< i, k >), k \geq 1$ ont le même temps de cycle moyen, et le même facteur de périodicité égal au ppcm des hauteurs des circuits critiques élémentaires de G.*

G quelconque

- G se décompose en composantes fortement connexes $C_1, \dots, C_r$. Chacune possède une valeur de circuit critique, notée $\alpha_1, \dots, \alpha_r$.
- On définit le graphe réduit R obtenu en contractant les sommets d'une même composante. C'est un graphe sans circuit, qui peut donc être numéroté de manière topologique.
- Supposons que les composantes fortement connexes sont numérotées dans cet ordre. On note $\Gamma_R^-(C_u)$ les prédécesseurs de la composante C_u dans le graphe réduit R.

Theorème 11. *Soit i une tâche de la composante C_u. La suite $t^{\sigma^*}(< i, k >), k \geq 1$ est K_u-périodique de période W_u avec*

$$\frac{W_u}{K_u} = \min(\alpha_u, \min_{v \leq u, C_v \in \Gamma_R^-(C_u)}, \frac{W_v}{K_v})$$

Observation 5. *On peut caractériser en temps polynomial le comportement de l'ordonnancement au plus tôt mais on ne sait pas le construire en temps polynomial (ni si cela est possible). Cela dépend de la longueur du régime transitoire et du facteur de périodicité. problèmes ouverts!!!*

Stabilité

Définition 5. *Le temps de réponse de l'ordonnancement σ pour l'itération $k \geq 1$ est égal à la durée de cette itération:*

$$\Delta_k^\sigma = \max_{i \in \mathcal{T}} t^\sigma(<i,k>) + p_i - \min_{i \in \mathcal{T}} t^\sigma(<i,k>)$$

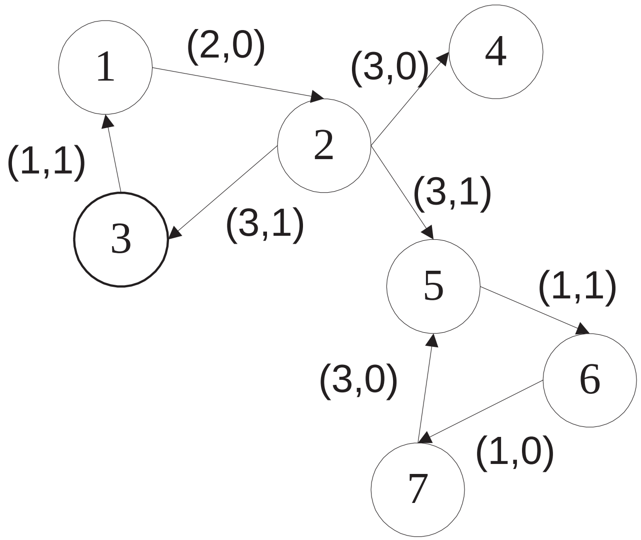
Définition 6. *Un ordonnancement infini σ est stable s'il existe une constante $S^\sigma \geq 0$ telle que*

$$\forall k \geq 1, \Delta_k^\sigma \leq S^\sigma$$

Theorème 12. *L'ordonnancement au plus tôt est stable si et seulement si toutes les tâches ont le même temps de cycle moyen $\alpha(G)$. Si le graphe n'est pas fortement connexe, alors il faut qu'il y ait un circuit critique de valeur $\alpha(G)$ dans toutes les composantes fortement connexe sans prédécesseur dans le graphe réduit R.*

Exemple

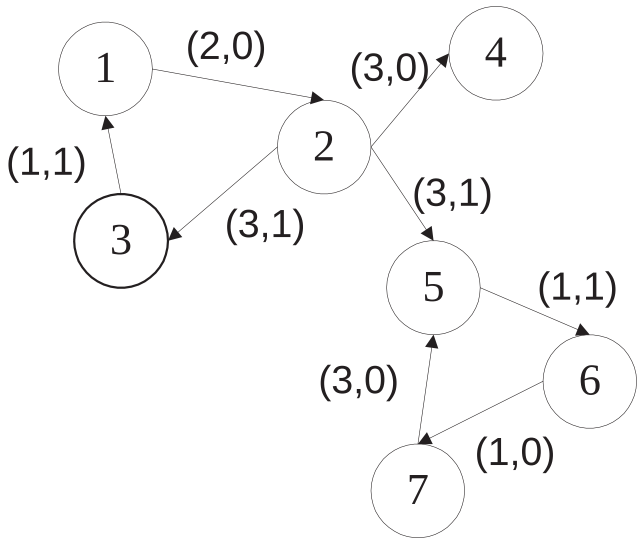
En supposant la tâche 4 de durée 1:



- Il y a 3 composantes fortement connexes: $C_1 = \{1, 2, 3\}, C_2 = \{4\}, C_3 = \{5, 6, 7\}$

Exemple

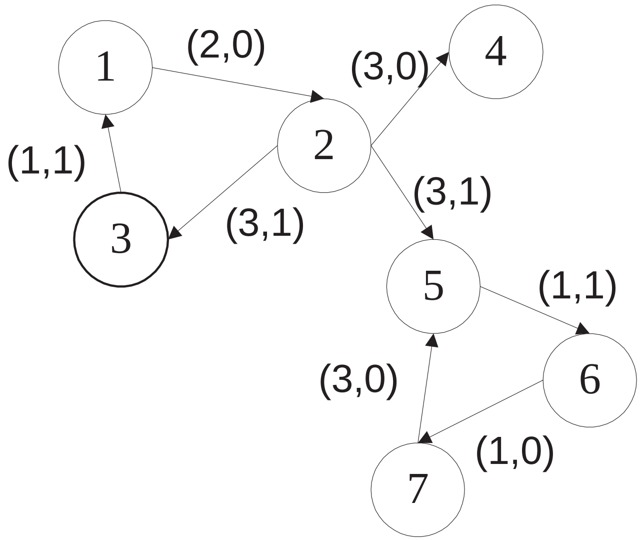
En supposant la tâche 4 de durée 1:



- Il y a 3 composantes fortement connexes: $C_1 = \{1, 2, 3\}, C_2 = \{4\}, C_3 = \{5, 6, 7\}$
- $\alpha(C_1) = 3, \alpha(C_2) = 1, \alpha(C_3) = 5$

Exemple

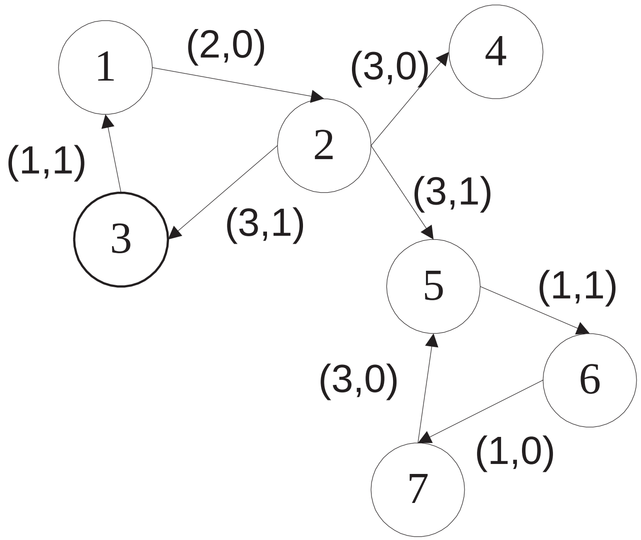
En supposant la tâche 4 de durée 1:



- Il y a 3 composantes fortement connexes: $C_1 = \{1, 2, 3\}, C_2 = \{4\}, C_3 = \{5, 6, 7\}$
- $\alpha(C_1) = 3, \alpha(C_2) = 1, \alpha(C_3) = 5$
- L'ordonnancement au plus tôt des tâches $i \leq 4$ est $2-$périodique de période 6 à partir d'un certain rang.

Exemple

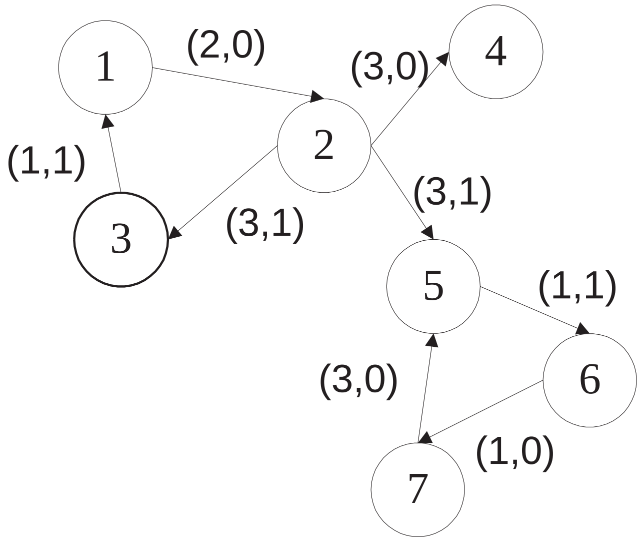
En supposant la tâche 4 de durée 1:



- Il y a 3 composantes fortement connexes: $C_1 = \{1, 2, 3\}, C_2 = \{4\}, C_3 = \{5, 6, 7\}$
- $\alpha(C_1) = 3, \alpha(C_2) = 1, \alpha(C_3) = 5$
- L'ordonnancement au plus tôt des tâches $i \leq 4$ est $2-$périodique de période 6 à partir d'un certain rang.
- L'ordonnancement au plus tôt des tâches $\{5, 6, 7\}$ est $1-$périodique de période 5 à partir d'un certain rang. arcs

Exemple

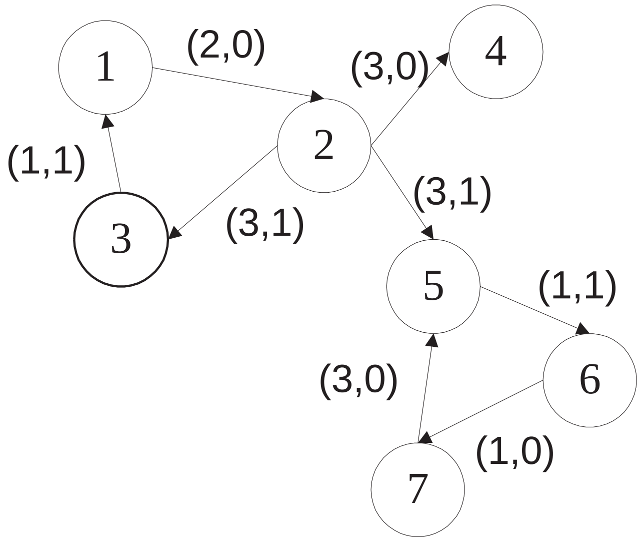
En supposant la tâche 4 de durée 1:



- Il y a 3 composantes fortement connexes: $C_1 = \{1,2,3\}, C_2 = \{4\}, C_3 = \{5,6,7\}$
- $\alpha(C_1) = 3, \alpha(C_2) = 1, \alpha(C_3) = 5$
- L'ordonnancement au plus tôt des tâches $i \leq 4$ est $2-$périodique de période 6 à partir d'un certain rang.
- L'ordonnancement au plus tôt des tâches $\{5,6,7\}$ est $1-$périodique de période 5 à partir d'un certain rang. arcs
- Cet ordonnancement est instable.

Exemple

En supposant la tâche 4 de durée 1:



- Il y a 3 composantes fortement connexes: $C_1 = \{1,2,3\}, C_2 = \{4\}, C_3 = \{5,6,7\}$
- $\alpha(C_1) = 3, \alpha(C_2) = 1, \alpha(C_3) = 5$
- L'ordonnancement au plus tôt des tâches $i \leq 4$ est $2-$périodique de période 6 à partir d'un certain rang.
- L'ordonnancement au plus tôt des tâches $\{5,6,7\}$ est $1-$périodique de période 5 à partir d'un certain rang. arcs
- Cet ordonnancement est instable.
- On peut construire un ordonnancement périodique (donc stable) de période 5.

Contraintes de resssource

En présence de telles contraintes, processeurs identiques ou spécialisés:

- Les ordonnancement périodiques sont-ils dominants?
- Peut-on construire des ordonnancement périodiques optimaux (complexité, algorithmes exacts et approchés)?
- Peut-on construire des politiques d'ordonnancement dynamique efficaces?

Exemple des processeurs identiques

Un ordonnancement σ affecte à chaque tâche $< i, k >, i \in \mathcal{T}, k \geq 1$

- une date d'exécution $t^\sigma(< i, k >)$
- un processeur $\pi^\sigma(< i, k >) \in \{1, \ldots, m\}$
- de sorte que chaque machine ne fasse qu'une tâche à la fois.

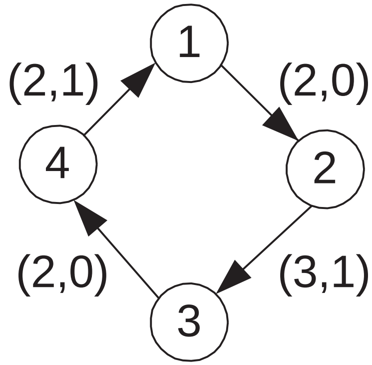
Quelques rappels sur les problèmes non-cycliques avec processeurs parallèles:

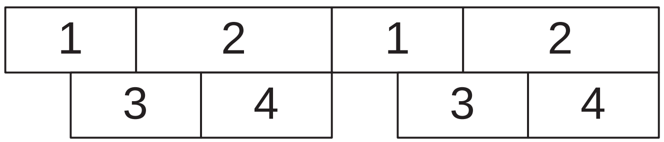
- $P2|prec, p_i = 1|C_{max}$ et $P|arbre, p_i = 1|C_{max}$ sont résolus en temps polynomial
- $P|prec, p_i = 1|C_{max}$ est NP complet au sens fort
- Les algorithmes de liste permettent de résoudre $P|prec|C_{max}$ avec une approximation relative $2 - \frac{1}{m}$ au mieux (Graham)

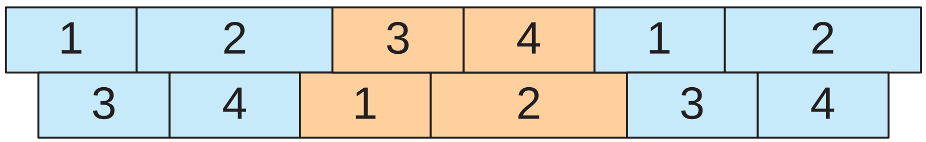
Périodicité?

Définition 7. *Un ordonnancement périodique est dit* *à affectation constante* *si*
$\forall k \geq 1, \pi^\sigma(< i, k >) = \pi^\sigma(< i, 1 >)$

Proposition 2. *(Munier)Les ordonnancements périodiques à affectation constante ne sont pas dominants parmi les ordonnacements périodiques*

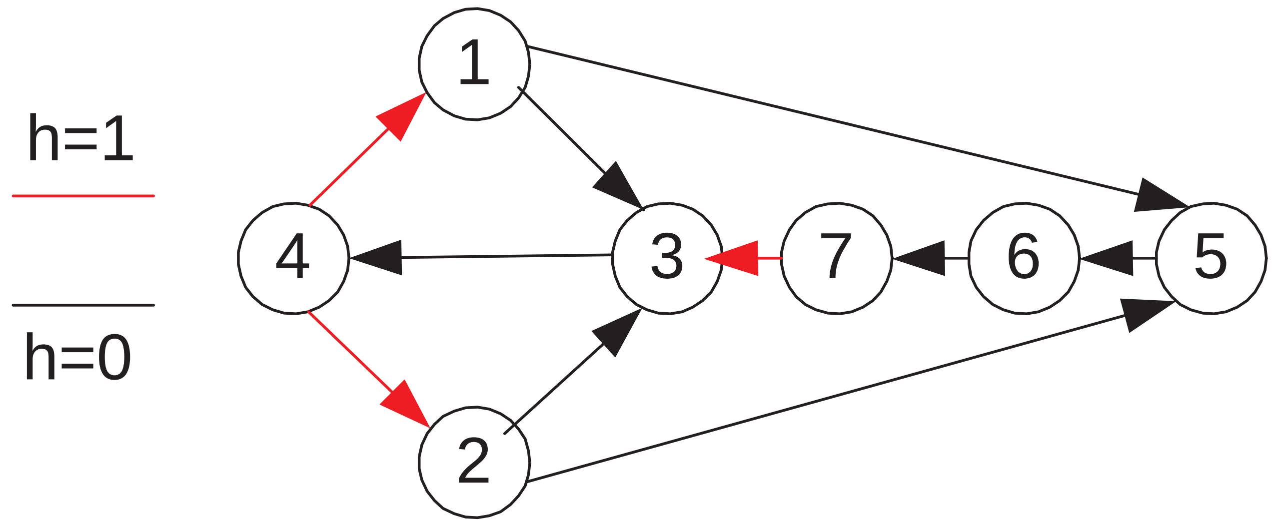






Non-dominance

Proposition 3. *(Munier) Les ordonnancements périodiques ne sont pas dominants pour le problème à machines identiques.*



1	3	4	1	7	3	4	1	3	4	1	7	3	4
2	5	6	2	5	6	7	2	5	6	2	5	6	7

Motif d'un ordonnancement périodique

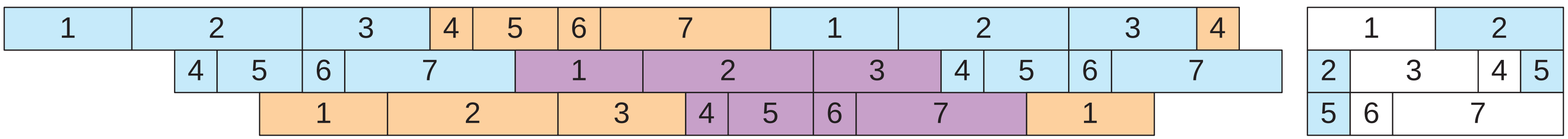
Soit σ un ordonnancement périodique.

Définition 8. *On appelle* Motif s^σ *de l'ordonnancement les dates d'exécution "fictives" calculées modulo la période:*

$$\forall i \in \mathcal{T}, \quad t_i^\sigma - t_1^\sigma = s_i^\sigma + w^\sigma \eta_i$$

avec $0 \leq s_i^\sigma < w^\sigma$

Le motif est l'ordonnancement sur une période commençant avec la tâche 1.



Proposition 4. *(Hanen-Munier) Un ordonnancement périodique σ satisfait les contraintes de ressource si et seulement si son motif les satisfait modulo w^σ*

Conditions d'existence d'un ordo à motif

Theorème 13. *(Hanen-Munier) Soit s un motif de période w. Il existe un ordonnancement périodique de motif s qui respecte les contraintes uniformes si et seulement si le graphe G muni de la valuation v suivante ne contient pas de circuit positif:*

$$\forall a, \quad i = b(a), j = e(a), \quad v(a) = \left\lceil \frac{s_i - s_j + p_i}{w} \right\rceil - H(a)$$

On peut alors construire η, potentiel sur (G, v)

Corollaire 4. *la plupart des algorithmes construisent un motif qui respecte les contraintes de ressource, puis un ordonnancement basé sur ce motif.*

Cas d'un graphe sans circuit

Si G est sans circuit hormis les boucles de non-réentrance.

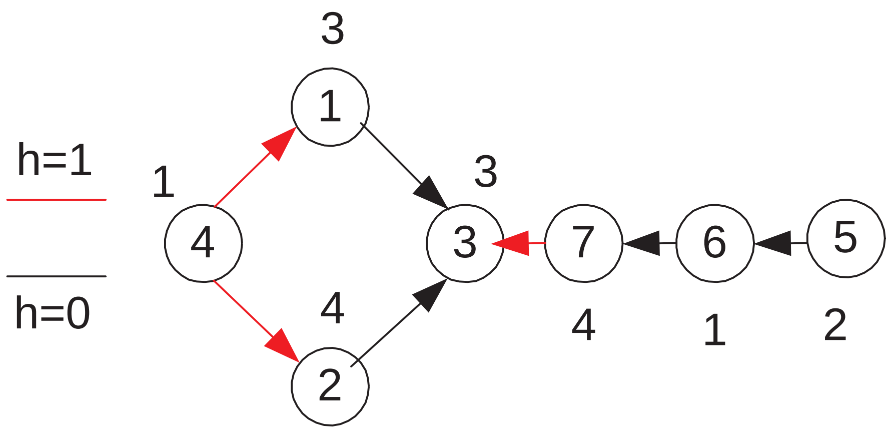
Proposition 5.

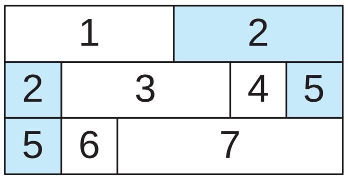
$$\alpha(\sigma^{opt}) \geq \max(\frac{\sum_{i\in\mathcal{T}} p_i}{m}, \max_{i\in\mathcal{T}} p_i) = B_m$$

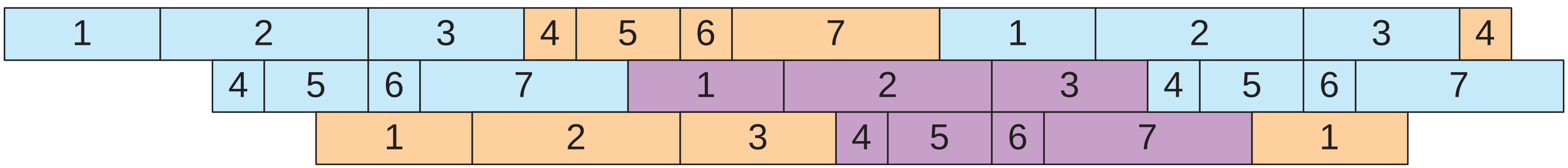
Theorème 14. *(Hanen Munier) Tout motif de période $w \geq B_m$ qui satisfait les contraintes de ressources induit un ordonnancement périodique qui satisfait les contraintes uniformes*

On peut construire un motif à l'aide de l'algorithme de Mac Naughton - ordonnancement de tâches préemptives

Exemple d'un graphe sans circuit







Cas d'un seul circuit

Si G est réduit à un circuit

Proposition 6.

$$\alpha(\sigma^{opt}) \geq \max(\frac{L(G)}{H(G)}, \frac{\sum_{i\in\mathcal{T}} p_i}{m}, \max_{i\in\mathcal{T}} p_i) = B_m$$

Proposition 7. *Le graphe développé $\mathcal{G}$ peut être partitionné en $H(G)$ chemins disjoints*

Theorème 15. *(Munier) Il existe un ordonnancement périodique de période B_m.*

Cas général

Theorème 16. *(Munier) Le problème de recherche d'un ordonnancement périodique de période minimale est NP-difficile.*

Preuve à partir du problème $P|prec|C_{max}$

Algorithme de Gasperoni-Schwiegelsohn

- Construire un ordonnancement périodique optimal sans les contraintes de ressources σ^{∞}.
- Retirer du graphe G les arcs $a = (i, j)$ qui "traversent" la frontière du motif: $s_i^{\sigma^{\infty}} + p_i > s_j^{\sigma^{\infty}} \quad \longrightarrow$ graphe G'
- Ordonnancer le graphe sans circuit G' sur m machines à l'aide d'un algorithme de liste.
- Utiliser cet ordonnancement comme motif d'un ordonnancement périodique sur m machines.

Theorème 17. *(Gasperoni-Schwiegelsohn) L'algorithme décrit ci dessus construit un ordonnancement σ tel que:*

$$\alpha(\sigma) \leq (2 - \frac{1}{m})\alpha(\sigma^{opt}) + \max_{i \in \mathcal{T}} p_i$$

Bibliographie

- Problèmes d'ordonnancement, J. Carlier, P. Chrétienne, Masson 1988
- Cyclic Scheduling on Parallel processors: An overview, C. Hanen et A. Munier, Scheduling Theory and its applications, John wiley and sons, 1995
- The cyclic scheduling problem with linear precedence constraints, A. Munier 1991.