Programmation linéaire en nombres entiers pour l'ordonnancement modulo sous contraintes de ressources.

* AYALA Maria.

* ARTIGUES Christian.

*** LAAS-CNRS; Université de Toulouse, France**
{mayala,artigues}@laas.fr

ROADEF 2009.

Plan.

1. Introduction.
2. Problème d'ordonnancement modulo.
3. PLNE pour l'ordonnancement modulo (état de l'art)
4. Génération de colonnes pour l'ordonnancement modulo
5. Résultats expérimentaux et conclusions

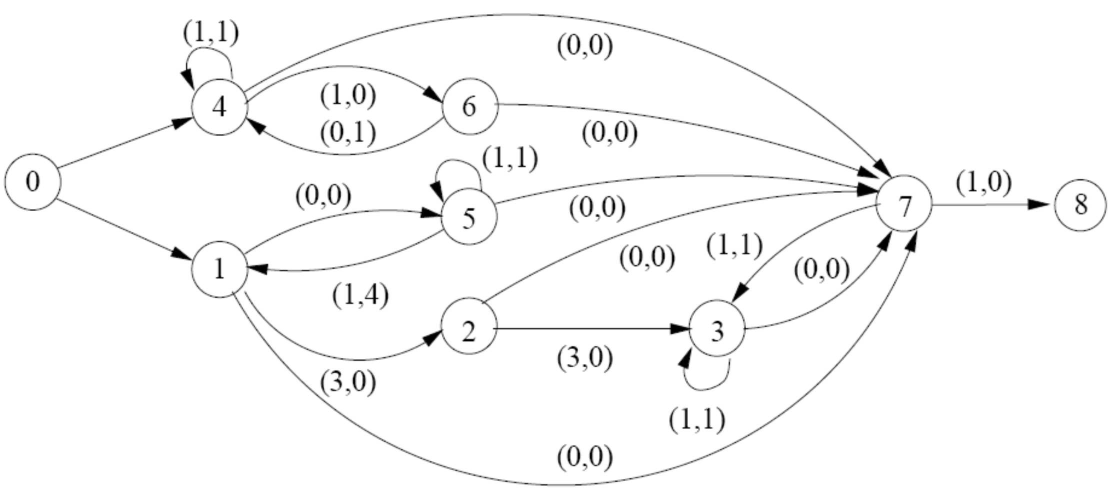
Introduction.

Exemple [1]

```
int
prod(int n, short a[], short b) {
    int s=0, i;
    for (i=0;i<n;i++) {
        s += a[i]*b;
    }
    return s;
}
```

```
L?__0_8:
LDH_1 g131 = 0, G127
MULL_2 g132 = G126, g131
ADD_3 G129 = G129, g132
ADD_4 G128 = G128, 1
ADD_5 G127 = G127, 2
CMPNE_6 b135 = G118, G128
BRF_7 b135, L?__0_8
```

Boucles



Proposer un ordonnancement des instructions
pour terminer le programme en un temps minimal

Introduction.

Proposer un ordonnancement des instructions pour terminer le programme en un temps minimal

Optimiser l'exécution des boucles

pipeline «logiciel»

probléme d'ordonnancement cyclique

aujourd'hui les compilateurs commerciaux utilisent le pipeline «logiciel»:

Ordonnancement modulo

Ordonnancement modulo.

- Ordonnancement Modulo = Ordonnancement cyclique 1- période

- Calculer un ordonnancement local puis le dupliquer à l'identique en espaçant les exécutions d'un intervalle constant appelé intervalle d'initiation (période λ).

- La minimisation de λ maximise le débit de l' ordonnancement

Ordonnancement modulo.

- Ensemble de tâches génériques $O = \{1, \ldots, n\}$
- Chaque tâche *i a une durée unitaire.*
- Chaque Tâche doit être exécutée plusieurs fois
- Soit $< i, k >$ *la* k- ième occurrence de la tâche *i*

- Ensemble de ressources s =1,…,m, chaque ressource a disponibilité B_s

  Chaque tâche Demandes b_i^s

- Date de debut

  $\sigma_i \longrightarrow$ ordonnancement local (période λ*) de la tâche générique i*

  $\sigma_i^k = \sigma_i + k\lambda \longrightarrow$ k-ième occurrence de la tâche *i*

Contraintes de dépendance.

- Graphe orienté $G = (O, E)$

  les sommets sont les tâches et les arcs sont caractérisés par deux fonctions :

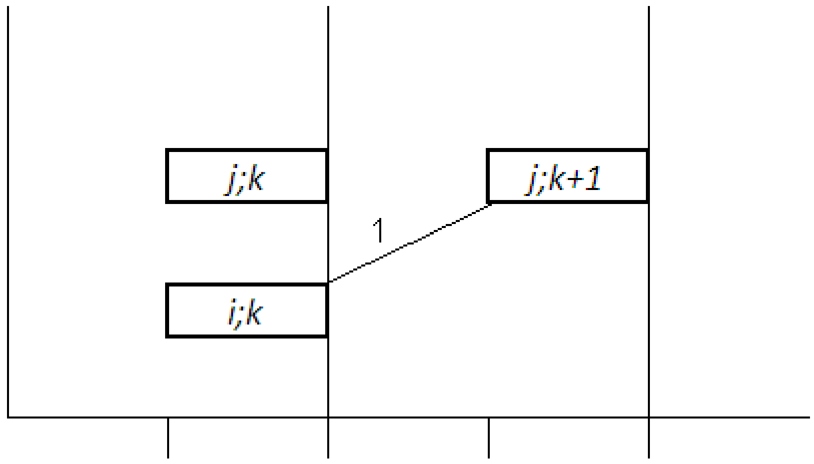
$\theta : O \rightarrow N$ Latence $\longrightarrow$ longueur de la dépendance

$\omega : O \rightarrow Z$ Distance $\longrightarrow$ nombre d'itérations qui séparent les instances de *i* et *j*

Contraintes de dépendance.

Soit (i, j) tel que $\theta_i^j = 1, \omega_i^j = 1$

$$\sigma_i^k = \sigma_i + k\lambda$$



$$\sigma_j^{\omega_i^j} \geq \sigma_i + \theta_i^j$$

$$\sigma_j^{k+\omega_i^j} \geq \sigma_i^k + \theta_i^j$$

$$\sigma_i + \theta_i^j - \omega_i^j \lambda \leq \sigma_j \quad \forall (i, j) \in E$$

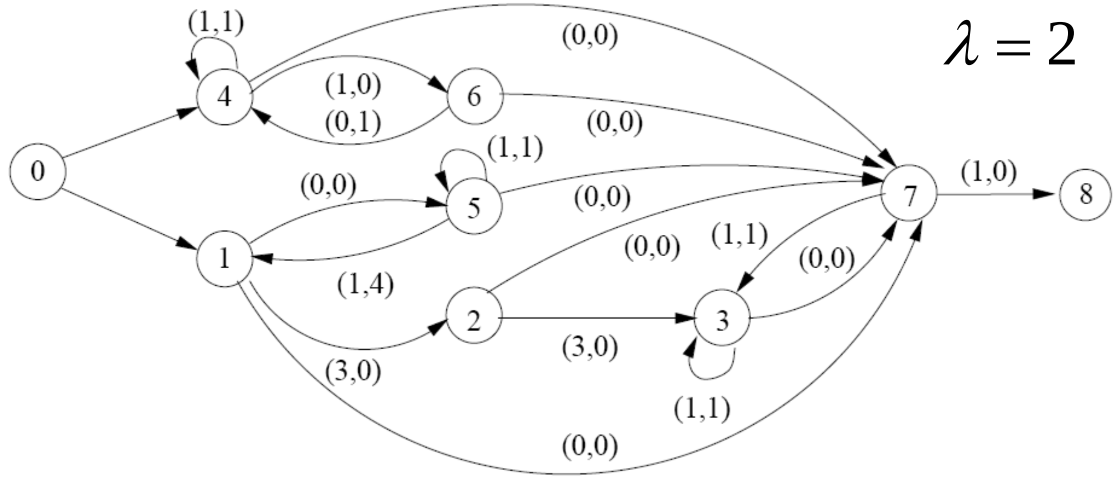
Ordonnancement Modulo sous contraintes de ressources

Resource	Available
ALU	4
MEM	1
CTL	1
ODD	2

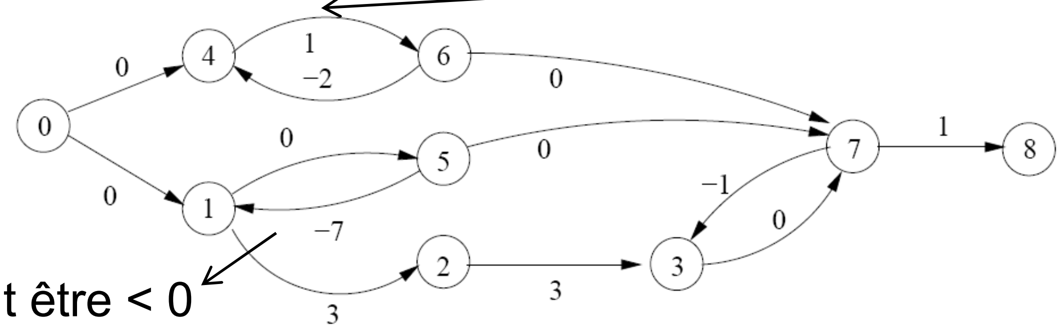
RESERVATION	ALU	MEM	CTL	ODD
ALU	1	0	0	0
ALUX	2	0	0	1
MUL	1	0	0	1
MULX	2	0	0	1
MEM	1	1	0	0
MEMX	2	1	0	1
CTL	1	0	1	1

RESERVATION = "type de tâche" qui définit la consommation des ressources

$\lambda = 2$

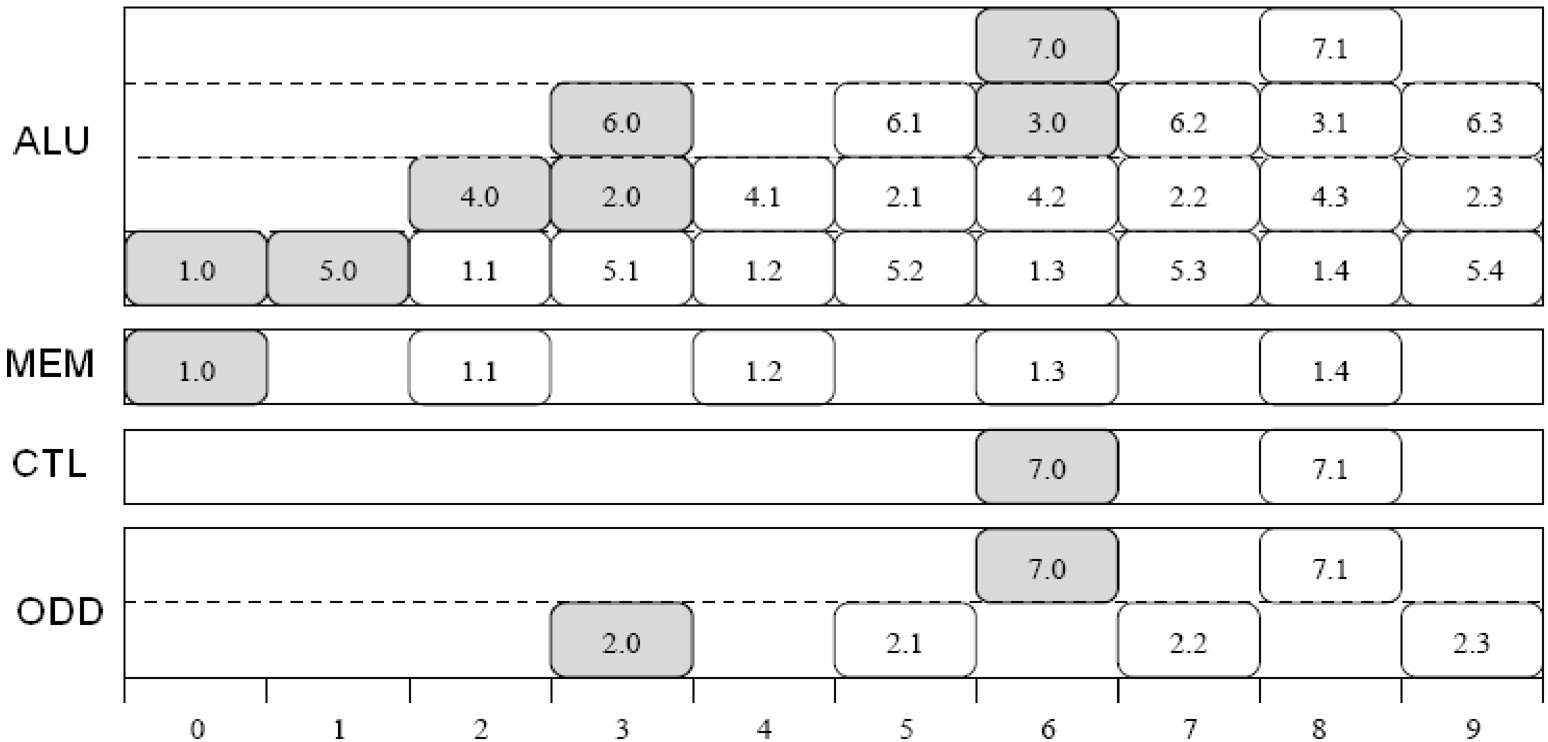


Si i=4 et j=6 $\theta_i^j = 1, \lambda\omega_i^j = 0$ et $\theta_i^j - \lambda\omega_i^j = 1$



$\theta_i^j - \lambda\omega_i^j$ Peut être < 0

problème acyclique avec délais minimaux (valeurs positives) et maximaux (valeurs négatives)


ALU
MEM
CTL
ODD
7.0
7.1
6.0
6.1
3.0
6.2
3.1
6.3
4.0
2.0
4.1
2.1
4.2
2.2
4.3
2.3
1.0
5.0
1.1
5.1
1.2
5.2
1.3
5.3
1.4
5.4
0
1
2
3
4
5
6
7
8
9

$$\lambda = 2$$

Formalisation du problème ordonnancement modulo sous contraintes de ressources.

$$\min \mathrm{Lex}(\lambda, \sum_{i=1}^{n} w_i \sigma_i)$$

$s.c$

$$\sigma_i + \theta_i^j - \omega_i^j \lambda \leq \sigma_j \quad \forall (i,j) \in E \qquad (1.7)$$

$$\sum_{O_i \in A(t)} b_i^s \leq B_s \quad \forall \mathrm{s} \in [1, \mathrm{m}], \forall t \in [0, \lambda[\qquad (1.8)$$

$$A(t) = \{O_i ; \sigma_i = t \bmod \lambda\}$$

l'objectif secondaire a pour but d'optimiser la gestion de registres virtuels

Formalisation du problème ordonnancement modulo sous contraintes de ressources.

Le problème de ordonnancement modulo peut être représenté de deux façons:

- Formulation directe: σ_i Début de la tâche générique *i* sur $[0, T-1]$

- Formulation décomposée : $\sigma_i = \lambda k_i + \tau_i$ Où:

$k_i \in \left\{0, \ldots, \left\lfloor \frac{T-1}{\lambda} \right\rfloor\right\}$ $i \in [1, n] \rightarrow$ est le numéro du cycle dans lequel chaque opération est placée.

$\tau_i = \sigma_i \bmod \lambda$ $\longrightarrow$ date de début dans l'intervalle [0,λ-1]

PLNE pour l'ordonnancement modulo.

Extensions de Christofides et al [1987] (pour le RCPSP acyclique)

1. Formulation decomposée par Eichenberger et al [1997]

2. Formulation directe Dupont-de-Dinechin et al [2004]

λ *constante*

Trouver par résolutions successives de PLNE la valeur minimale de λ

Définition d'un objectif secondaire lié à l'optimisation de l'utilisation des registres

Formulation décomposée.
Eichenberger et al [1997]

variables binaires :

$$z_i^\tau \in \{0,1\}, \qquad i = 1,\ldots,n \qquad \tau \in [0, \lambda - 1]$$ Tel que $$\tau_i = \sum_{\tau=0}^{\lambda-1} \tau z_i^\tau$$

$$\min \sum_{i=1}^{n} w_i \sigma_i = \min w_i (\lambda k_i + \tau_i) \qquad k_i \in N, \quad \tau_i \in [0, \lambda \text{-} 1]$$

Contraintes d'affectation $$\sum_{\tau=0}^{\lambda-1} z_i^\tau = 1$$

Contraintes de dépendances $$\sum_{\tau=0}^{\lambda-1} \tau z_i^\tau + k_i \lambda + \theta_i^j - \lambda \omega_i^j \le \sum_{\tau=0}^{\lambda-1} \tau z_j^\tau + k_j \lambda$$

Contraintes ressources $$\sum_{i=1}^{n} z_i^\tau b_i^s \le B_s, \ \forall s \in [1, m], \tau \in [0, \lambda)$$

Formulation décomposeé (version structurée). Eichenberger et al [1997]

Les nouvelles contraintes de dépendances :

$$\sum_{x=\tau}^{\lambda-1} z_x^i - \sum_{x=0}^{(\tau+\theta_i^j-1)\bmod\lambda} z_x^j + k_i - k_j \leq \omega_i^j - \left\lfloor \frac{\tau+\theta_i^j-1}{\lambda} \right\rfloor + 1$$

Formulation finale des contraintes de dépendances structurées.

Relaxation de PL meilleure avec les contraintes « structurées ».

Formulation directe.
Dupont-de-Dinechin et al [2004]

Variables x_i^t 0 - 1 tel que $\sigma_i = \sum_{t=0}^{T-1} t x_i^t \qquad i = 1, \dots, n$

$$\min \sum_{i=1}^{n} w_i \sigma_i = \min \sum_{i=1}^{n} w_i \left(\sum_{t=0}^{T} t x_i^t \right) \qquad i = 1, \dots, n$$

$$s.t$$

$$\sum_{t=0}^{T} x_i^t = 1 \qquad i = 1, \dots, n$$

$$\sum_{t=0}^{T} t x_i^t + \theta_i^j - \lambda \omega_i^j \leq \sum_{t=0}^{T} t x_j^t \qquad (i, j) \in E$$

$$\sum_{i=1}^{n} \sum_{k=0}^{\left\lfloor \frac{T}{\lambda} \right\rfloor} x_i^{\tau + k\lambda} b_i^s \leq B_s \quad \forall\, s \in [1, \dots, m], \tau \in [0, \lambda - 1]$$

Formulation directe. (version désagrégée)

Dupont-de-Dinechin et al [2004]

Il est possible d'introduire des contraintes de dépendances désagrégées pour la formulation indexée du temps

Remplaçant

$$\sum_{t=0}^{T} t x_i^t + \theta_i^j - \lambda \omega_i^j \leq \sum_{t=0}^{T} t x_j^t \qquad (i, j) \in E$$

par

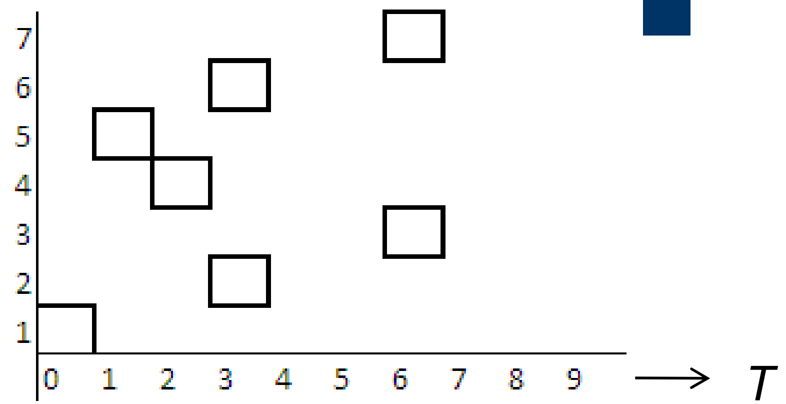
$$\sum_{s=t}^{T} x_i^s + \sum_{s=0}^{t+\theta_i^j - \lambda \omega_i^j - 1} x_j^s \leq 1$$

La formulation avec contraintes de dépendances désagrégées donne une meilleure relaxation de PL (cf Christofides et al 1987).

Comparaison des formulations.

- Formulation directe:

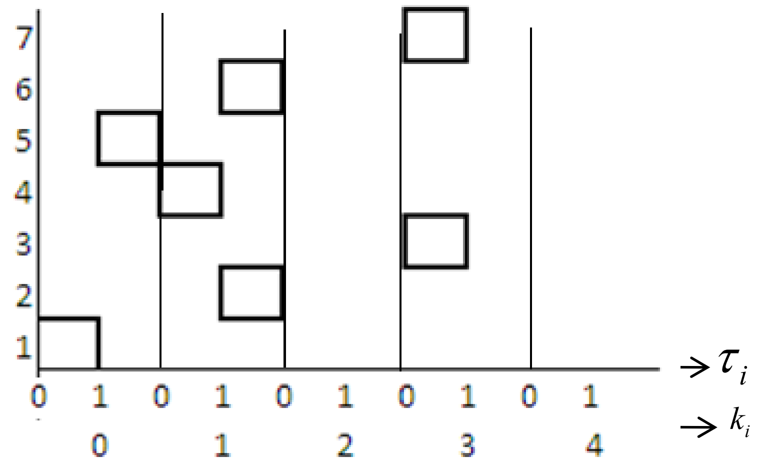
$$n \times t$$



- Formulation décomposée :

$n \times \lambda$ variables binaires +

n variables entières



Formulation Mingozzi et al [1995].

basée sur les ensembles de tâche réalisables pour le RCPSP acyclique

Sous - ensembles réalisables de tâches en cours d'exécution à l'instant *t* :

1. Les contraintes de ressource sont satisfaites
2. Il n'existe pas contraintes de dependance entre (i,j).

Un sequence de sous-ensembles réalisables représente une solution si elle satisfait les conditions:

- ❑ Chaque tâche est exécutée sans interruption.
- ❑ Les date de debut des tâches satisfaitont les contraintes de dependance.

Example.

Nombre d'ensembles maximum $2^8 = 256$

λ= 2; tâches = 8; dependances = 14; sous-ensambles=61;

[1,0,0,0,0,0,0,0], [0,2,0,0,0,0,0,0], [0,0,3,0,0,0,0,0], [0,0,0,4,0,0,0,0],[0,0,0,0,5,0,0,0]
[0,0,0,0,0,6,0,0], [0,0,0,0,0,0,7,0], [1,0,3,0,0,0,0,0], [1,0,0,4,0,0,0,0], [1,0,0,0,0,5,0,0]
[1,0,0,0,0,6,0,0], [0,2,3,0,0,0,0,0], [0,2,0,4,0,0,0,0], [0,2,0,0,5,0,0,0], [0,2,0,0,0,6,0,0]
[0,0,3,4,0,0,0,0], [0,0,3,0,5,0,0,0], [0,0,3,0,0,6,0,0], [0,0,3,0,0,0,7,0], [0,0,0,4,5,0,0,0]
[0,0,0,4,0,6,0,0], [0,0,0,4,0,0,7,0], [0,0,0,0,5,6,0,0], [0,0,0,0,5,0,7,0], [0,0,0,0,0,6,7,0]
[1,0,3,4,0,0,0,0], [1,0,3,0,5,0,0,0], [1,0,3,0,0,6,0,0], [1,0,0,4,5,0,0,0], [1,0,0,4,0,6,0,0]
[1,0,0,0,5,6,0,0], [1,0,0,0,5,0,7,0], [0,2,3,4,0,0,0,0], [0,2,3,0,5,0,0,0], [0,2,3,0,0,6,0,0]
[0,2,0,4,5,0,0,0], [0,2,0,4,0,6,0,0], [0,2,0,0,5,6,0,0], [0,0,3,4,5,0,0,0], [0,0,3,4,0,6,0,0]
[0,0,3,4,0,0,7,0], [0,0,3,0,5,6,0,0], [0,0,3,0,5,0,7,0], [0,0,3,0,0,6,7,0], [0,0,0,4,5,6,0,0],
[0,0,0,4,5,0,7,0], [0,0,0,4,0,6,7,0], [0,0,0,0,5,6,7,0], [1,0,3,4,5,0,0,0], [1,0,3,4,0,6,0,0]
[1,0,3,0,5,6,0,0], [1,0,0,4,5,6,0,0], [0,2,3,4,5,0,0,0], [0,2,3,4,5,6,0,0], [0,2,3,0,5,6,7,0]
[0,2,0,4,5,6,7,0], [0,0,3,4,5,6,0,0], [0,0,3,4,5,0,7,0], [0,0,3,4,0,6,7,0], [0,0,3,4,5,6,7,0]
[0,0,0,4,5,6,7,0]

Formulation Mingozzi.

$y_l^t \longrightarrow$ variable binaire (0-1) égale à 1 ssi toutes les tâches du sous-ensemble réalisables R_l sont en cours d'exécution à la date t.

$x_i^t \longrightarrow$ variable binaire (0-1) égale à 1 ssi la tâche i commence à la date t.

Extension pour l'ordonnancement modulo :

$y_l^\tau = 1, \qquad \tau \in [0, \lambda - 1]$ si les tâches sont en cours d'exécution à l'instant τ de la période générique

variables binaires $\left\{ \lambda \times \text{sous - ensembles réalisables} \right.$

Extension pour l'ordonnancement Modulo (formulation directe).

Soit
R = ensemble de sous-ensembles réalisables

$$\min \sum_{i=1}^{n} w_i \sum_{t=0}^{T} t x_i^t + 1$$

$$s.t$$

$$\sum_{t=0}^{T} x_i^t = 1 \qquad \forall i = 1,\dots, n$$

$$\sum_{l \in R} \left(\sum_{\tau=0}^{\lambda-1} a_{il} y_l^\tau \right) = 1 \qquad \forall i = 1,\dots, n$$

$$\sum_{l \in R} y_l^\tau \leq 1 \qquad \tau \in [0, \lambda - 1]$$

$$\sum_{t=0}^{T} t x_i^t + \theta_i^j - \lambda \omega_i^j \leq \sum_{t=0}^{T} t x_j^t \qquad (i,j) \in E$$

$$\sum_{l \in R} a_{il} y_l^\tau = \sum_{t=0}^{T-1;(t/\lambda)=\tau} x_i^t \qquad \forall i = 1,\dots, n, \tau \in [0, \lambda - 1]$$

Extension pour l'ordonnancement Modulo (formulation decomposée).

$$\min \sum_{i=1}^{n} w_i (k_i \lambda + \sum_{\tau=0}^{\lambda-1} r \sum_{l \in R} y_l^{\tau})$$

sc

$$\sum_{l \in R} (\sum_{r=0}^{\lambda-1} a_{il} y_l^{\tau}) = 1,\ \ i = 1,\ldots,n$$

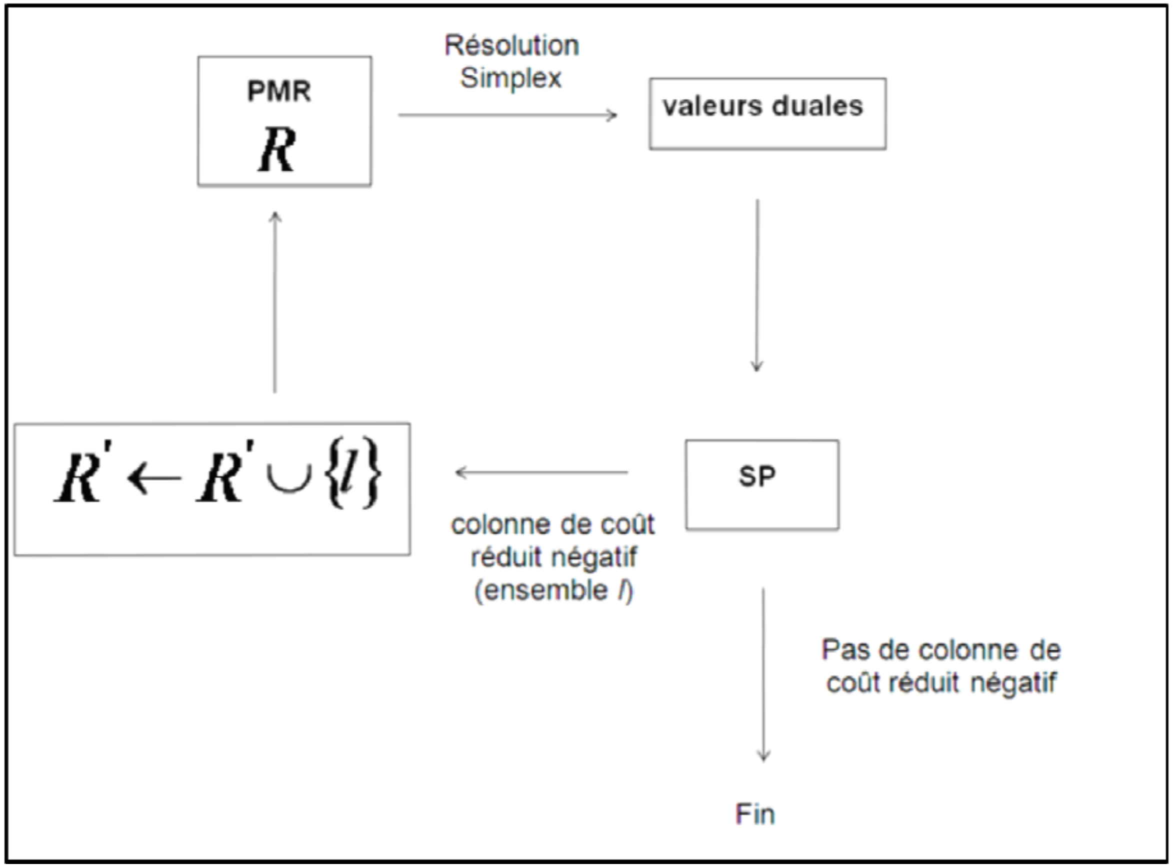
$$\sum_{l \in R} y_l^{\tau} \leq 1,\ \ r \in [0, \lambda - 1]$$

$$\sum_{\tau=0}^{\lambda-1} \tau (\sum_{l \in R} a_{jl} y_l^{\tau} - \sum_{l \in R} a_{il} y_l^{\tau}) + (k_j - k_i)\lambda \geq \theta_i^j - \omega_i^j \lambda \quad \forall (i, j) \in E$$

Résolution de la relaxation continue des PLNE

R = Toutes les ensembles possibles réalisables

$R' \subset R$



Problème dual (formulation directe).

.

$$\max \sum_{i=1}^{n} \mu_i + \sum_{i=1}^{n} a_{il}\sigma_i + \beta_\tau + (\lambda\omega_i^j - \theta_i^j)\gamma_i^j$$

$s.c$

$$\mu_i - \sum_{(i,j)\in E} t\gamma_i^j + \sum_{(j,i)\in E} t\gamma_j^i + \sum_{t\in[o,\lambda-1];t \bmod \lambda=r} \delta_i^\tau \leq tw_i \qquad i=1,\dots,n, \tau\in[o,\lambda-1]$$

$$\sum_{i=1}^{n} a_{il}\sigma_i - \beta_\tau + \sum_{i=1}^{n} a_{il}\delta_i^\tau \leq 0 \qquad \tau\in[o,\lambda-1], l \text{ en } \Omega.$$

Sous-problème

$$\sum_{i=1}^{n} a_{il}\sigma_i - \beta_\tau + \sum_{i=1}^{n} a_{il}\delta_i^\tau > 0 \quad \Longrightarrow \quad \sum_{i=1}^{n} (\sigma_i + \delta_i^\tau) a_{il} > \beta_\tau$$

For $\tau = 0, \ldots, \lambda - 1$

On a λ sous-problèmes

$$\left\{ \begin{array}{l} \max \sum_{i=1}^{n} (\sigma_i - \delta_i^\tau) a_i \\ s.c \\ \sum_{i=1}^{n} a_i b_i^s \leq B_s \end{array} \right.$$

Résultats expérimentaux et conclusions

Instance fft32x32s provenant du Compilateur ST200 de l'entreprise Stmicroelectronics.

85 tâches, 543 dépendances

Ressource	Disponibilité
ISSUE	4
MEM	1
CTL	1
ODD	2
EVEN	2
LANEO	2

	RESSOURCES					
RESERVATION	ISSUE	MEM	CTL	ODD	EVEN	LANEO
ALL	4	0	0	0	0	0
ALU	1	0	0	0	0	0
ALUX	2	0	0	1	1	0
CTL	1	0	1	1	1	0
ODD	1	0	0	1	0	0
ODDX	2	0	0	1	1	0
MEM	1	1	0	0	0	0
MEMX	2	1	0	1	1	0
PSW	0	0	0	0	0	0
EVEN	1	1	1	1	1	0

RESERVATION = "type de tâche" qui définit la consommation des ressources

Résultats expérimentaux et conclusions

	Formulation decomposée relachée	Formulation decomposée estructurée relachée	Formulation decomposée entière	Formulation decomposée estructurée entière
λ = 22	Cmax = 17,90	Cmax = 18,86	Cmax = 22 (300 seg)	Cmax = 22 (300 seg)

	Formulation directe agrégée relachée	Formulation directe relachée	Formulation directe agrégée entière (300 seg)	Formulation directe entière (300 seg)
λ = 22	Cmax = 17,90	Cmax = 18,86	Cmax = 22 (4526 seg)	Cmax = 22 (4395 seg)

λ = 20 : tous les PL sont infaisables

une borne inférieure: λ = 21
une borne supérieure: λ = 22

Suite des travaux.

- Démontrer l'équivalence entre les PLNE directs et décomposés
- Tester la qualité de la relaxation par génération de colonnes
- Proposer des méthodes approchées basées sur la PLNE et/ou spécifiques

Merci pour votre attention