L'impact des délais de communications hiérarchiques sur l'approximation et la complexité des problèmes d'ordonnancement

Rodolphe Giroudeau
Sous la direction du Professeur
Evripidis Bampis (LaMI, Université d'Évry Val d'Essonne)

Université d'Évry-Val d'Essonne

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
LaMI

Plan

- Motivations et exemple,
- Résultat de non-approximabilité,
- Résultats d'approximabilité sur plusieurs problèmes,
- Conclusions et perspectives.

Introduction

Motivation :

Évolution des architectures parallèles : architectures à base de modules de processeurs

Exemples :

- processeurs connectés par des bus hiérarchiques
- bi-processeurs connectés avec des switches myrinet

Nécessité de prendre en compte la notion de communications hiérarchiques

Communications intra-module (notée ϵ_{ij}) et extra-module (notée c_{ij})

Comment

$\Rightarrow$ Extension du modèle classique d'ordonnancement avec communications

Entrée du problème :

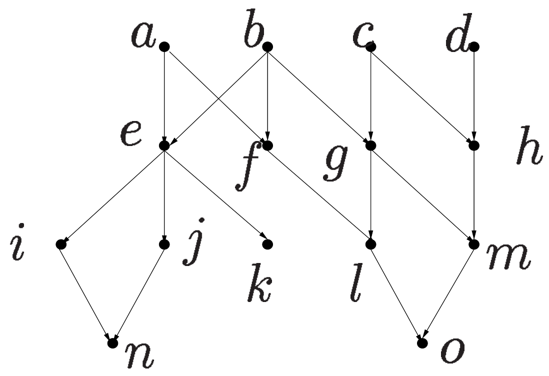
- $G = (V, E)$
- V : ensemble de tâches et $\forall i \in V, p_i$: durée d'exécution
- E : ensemble d'arcs

Communications :	Modèle classique	Modèle hiérarchique
	$\forall (i,j) \in E,\ c_{ij}$	$\forall (i,j) \in E,\ (c_{ij}, \epsilon_{ij})$ tel que $c_{ij} \geq \epsilon_{ij}$
		Si $c_{ij} = \epsilon_{ij} \Rightarrow$ modèle classique

Objectif : minimiser la durée totale de l'ordonnancement C_{max}

Problèmes $\mathcal{NP}$-difficiles

Exemple



$(c_{ij}, \epsilon_{ij}) = (1, 0)$

$p_i = 1$

	0	1	2	3	4	5
π_0^0	a	e	i	n		
π_1^0	b	f	j			
π_0^1	c	h		k		
π_1^1	d					
π_0^2			g	l	o	
π_1^2				m		

Ordonnancement du graphe avec deux processeurs par module

$c_{ij} = 1$

$p_i = 1$

	0	1	2	3	4	5	6	7
π_0	a				k			
π_1	b							
π_2			e	i				
π_3			f					
π_4	c				j	n		
π_5	d				m		o	
π_6			g		l			
π_7			h					

Ordonnancement du graphe avec un processeur par module

L'objectif de notre étude

Mesurer l'impact de l'introduction des délais de communications hiérarchiques sur la complexité et l'approximation des problèmes d'ordonnancement

$$\rho^h = \max_{G} \frac{C^h_{max}(G)}{C^{opt}_{max}(G)}.$$

Dans un premier temps : cas simplifiés

- Résultat de non-approximabilité
- Résultats d'approximabilité

Résultats négatifs

Modèle classique

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = 1$
- Nombre de processeurs suffisant

$\nexists$ ρ-approximation avec $\rho < \frac{7}{6}$, sauf si $\mathcal{P} = \mathcal{NP}$ [Hoogeveen,Lenstra,Veltman 94]

Modèle hiérarchique

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = 1$ et $\epsilon_{ij} = 0$
- Nombre de modules suffisant, 2 processeurs identiques par module

$\nexists$ ρ-approximation avec $\rho < \frac{5}{4}$, sauf si $\mathcal{P} = \mathcal{NP}$ [Bampis,Giroudeau,König 99]

Résultats positifs (1)

Modèle classique

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = 1$
- Nombre de processeurs suffisant

algorithme $\frac{4}{3}$-approché [Munier,König 97]

Modèle hiérarchique

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = 1$ et $\epsilon_{ij} = 0$
- Nombre de modules suffisant, 2 processeurs identiques par module

algorithme $\frac{8}{5}$-approché [Bampis,Giroudeau,König 99]

Idée de l'algorithme

1. Formulation du problème par un programme linéaire en nombre entiers
2. Relaxation des contraintes $\rightarrow$ programme linéaire
3. Résolution du programme linéaire $\rightarrow B_{inf}$ pour C_{max}
4. *Étape 1:* Étape d'arrondi sur les variables
5. *Étape 2:* Étape d'arrondi réalisable pour obtenir un ordonnancement réalisable
6. Comparaison entre C_{max} et B_{inf}

Formulation

$$
\left\{
\begin{array}{ll}
 & \min C_{max} \\
\forall (i,j) \in E, & x_{ij} \in \{0,1\} \\
\forall i \in V, & t_i \geq 0 \\
\forall (i,j) \in E, & t_i + p_i + x_{ij} \leq t_j \\
\forall (i,j) \in E - U, & \displaystyle\sum_{j \in \Gamma^+(i)} x_{ij} \geq |\Gamma^+(i)| - 2 \\
\forall (i,j) \in E - Z, & \displaystyle\sum_{j \in \Gamma^-(i)} x_{ji} \geq |\Gamma^-(i)| - 2 \\
\forall i,j,k,l,m \in V, & x_{ji} + x_{jk} + x_{lk} + x_{lm} \geq 1 \\
(j,i),(j,k),(l,k),(l,m) \in E, & \\
\forall i,j,k,l,m \in V, & x_{ij} + x_{kj} + x_{kl} + x_{ml} \geq 1 \\
(i,j),(k,j),(k,l),(m,l) \in E, & \\
\forall i \in V, & t_i + p_i \leq C_{max}
\end{array}
\right.
$$

Description

-Tout ordonnancement doit satisfaire les contraintes du programme linéaire en nombre entiers.

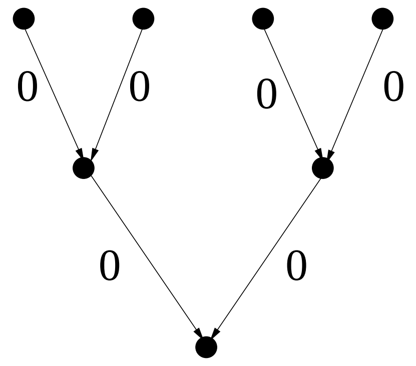
Lemme : Chaque tâche $i \in V$ possède au plus deux tâches successeurs (resp. prédécesseurs) tel que $e_{ij} < 0.25$ (resp. $e_{ji} < 0.25$).

Étape 1 [Arrondi]:

Si $e_{ij} < 0.25 \Rightarrow x_{ij} = 0$ (noté $0 - arc$) sinon $x_{ij} = 1$, $(1 - arc)$

Description

Un ordonnancement optimal issu du programme linéaire en nombre entiers $\not\Rightarrow$ ordonnancement réalisable



$\Rightarrow$ Seconde étape d'arrondi

Description

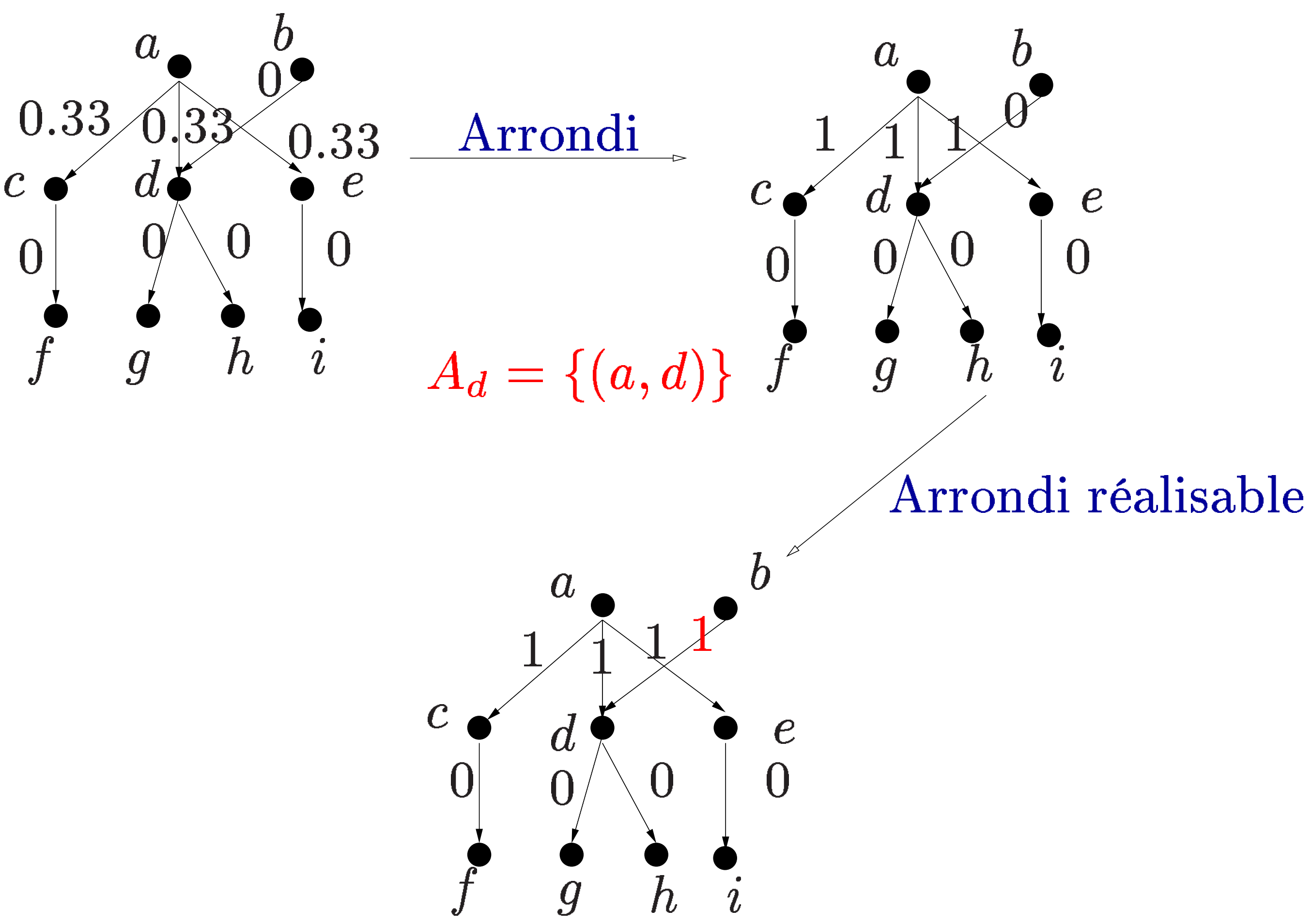
Arrondi réalisable : Soit A_i, l'ensemble des arcs qui sont éléments d'un chemin critique ayant pour sommet terminal la tâche i.

1. Si $\Gamma^{-}(i) = \varnothing$, nous gardons inchangé les valeurs des variables x_{ij} obtenues dans l'*Étape 1*,

2. Si *0-arc* $\in A_i$, $\forall k, (i,k) \in E, x_{ik} = 1$,

3. Si *0-arc* $\notin A_i$, alors la valeur des arcs sortants x_{ij} reste inchangée comme dans l'*Étape 1* & tous les *0-arcs* entrants sont transformés en *1-arcs*.

Exemple



a
b
0.33
0.33
0.33
0
c
d
e
0
0
0
0
f
g
h
i
Arrondi
$A_d = \{(a,d)\}$
a
b
1
1
1
0
c
d
e
0
0
0
0
f
g
h
i
Arrondi réalisable
a
b
1
1
1
1
c
d
e
0
0
0
0
f
g
h
i

Performance relative

Performance relative de l'heuristique

-Soit t_i^h le début d'exécution de la tâche i donné par l'heuristique h.

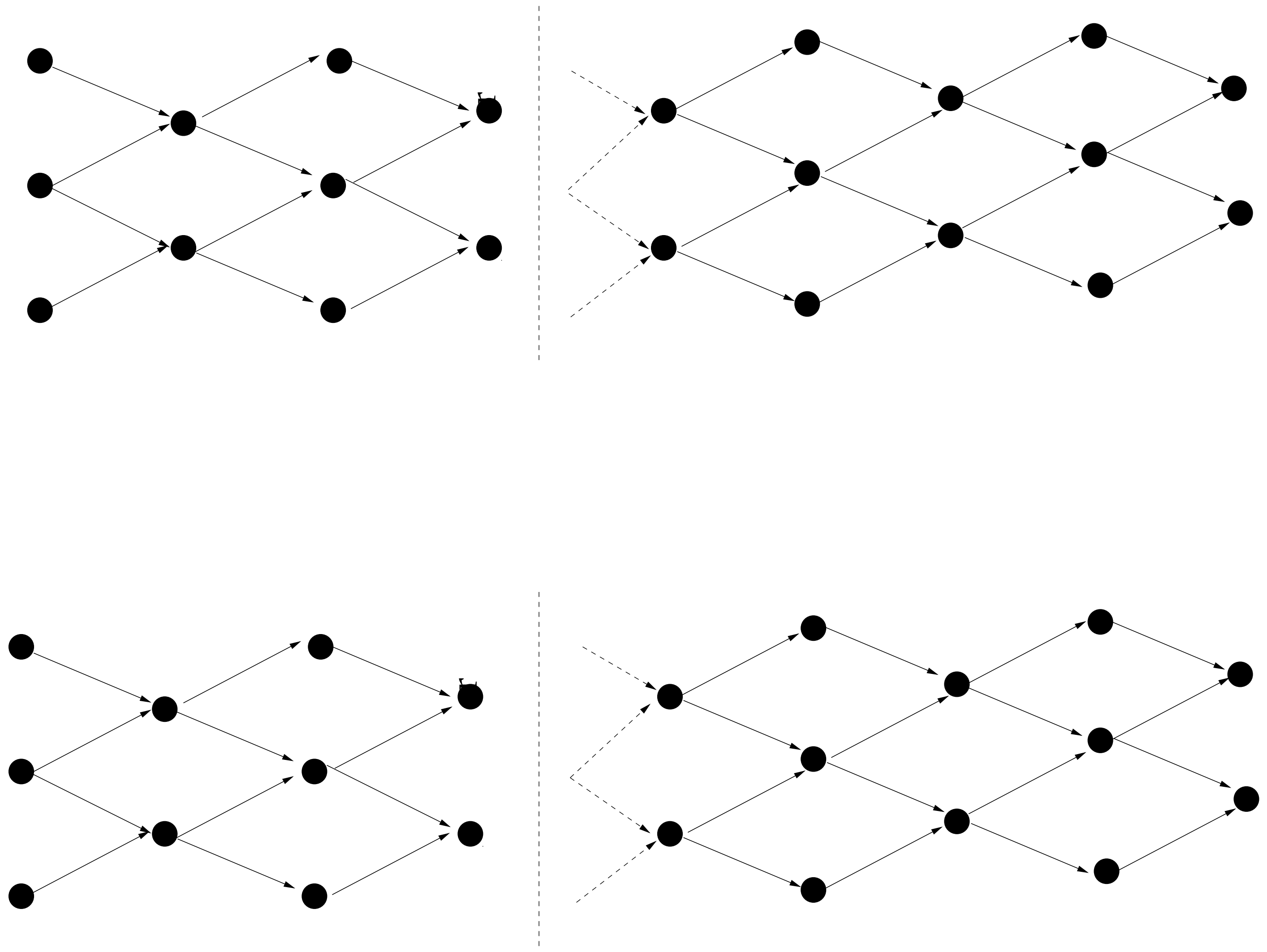
-Soit t_i^* le début d'exécution de la tâche i donné par le programme linéaire.

Lemme : Pour chaque tâche $i \in V, t_i^h \leq \frac{8}{5} t_i^*$.

Preuve : La preuve se fait par induction sur les tâches i.

1. A_i contient un $0 - arc$ noté (j, i) tel que $e_{ji} < 0.25$.
2. A_i contient un $1 - arc$ noté (j, i) tel que $e_{ji} \geq 0.25$.
3. A_i contient un $1 - arc$ noté (j, i) tel que $e_{ji} < 0.25$.

Borne supérieure



Extensions

→ algorithme $\frac{4m}{2m+1}$-approché pour $\bar{P}(Pm)|prec, p_i \geq 1, (c_{ij}, \epsilon_{ij}) = (1,0)|C_{max}$ où m est une constante fixée

→ algorithme $\frac{12(\Phi+1)}{12\Phi+1}$-approché pour $\bar{P}(P2)|prec, p_i, (c_{ij}, 0); \Phi \geq 1|C_{max}$ où $\Phi = \frac{\min\limits_{i \in V} p_i}{\max\limits_{(k,j) \in E} c_{kj}} \geq 1$ [Bampis,Giroudeau,Kononov 00]

Résultats positifs (2) en présence de la duplication

Modèle classique

- $\forall j \in V, \min_{i \in \Gamma^-(j)} \{p_i\} \geq \max_{i \in \Gamma^-(j)} \{c_{ij}\}$
- Nombre de processeurs suffisant, duplication

algorithme polynomial [Colin,Chrétienne 91]

Modèle hiérarchique

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = 1$ et $\epsilon_{ij} = 0$
- Nombre de modules suffisant, 2 processeurs identiques par module, duplication

algorithme polynomial [Bampis,Giroudeau,König 99]

Idée de l'algorithme

1. La première étape de l'algorithme calcule pour chaque tâche (et ses copies) sa date de début d'exécution au plus tôt et construit un graphe critique.

2. La seconde étape utilise le graphe critique et la duplication des tâches pour construire un ordonnancement optimal.

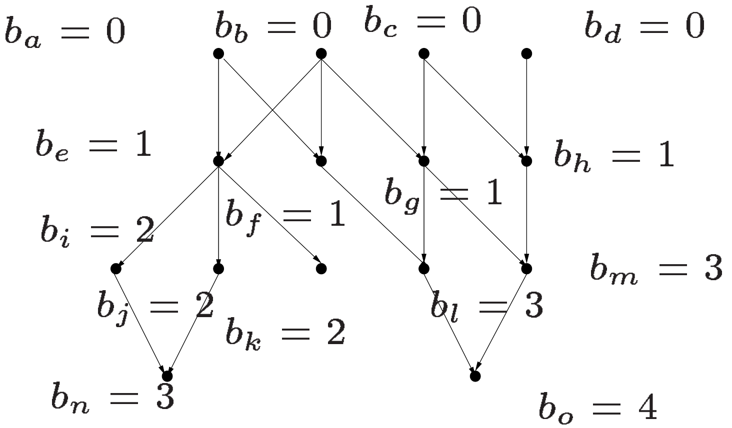
Algorithme

Définition: Une tâche i est un prédécesseur critique de j si l'arc $(i, j) \in E$ et $b_j = b_i + 1$.

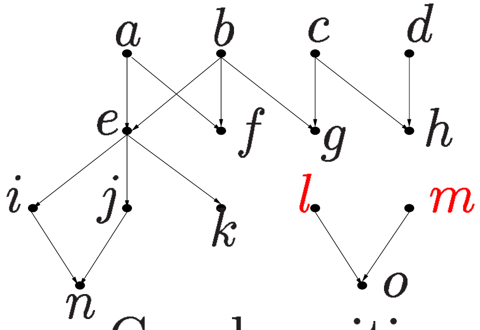
1. La borne inférieure des dates de début d'exécution :
   - Pour chaque source i de G, $b_i = 0$.
   - Si dans le graphe induit par les les prédécesseurs critiques de i, il existe au moins 3 prédécesseurs qui sont à la même distance l et ayant la date de début d'exécution au plus tôt maximale alors une communication est introduite entre i et ses prédécesseurs critiques.
2. Affecter chaque sous-graphe à un cluster différent et exécuté chaque copie des tâches à leur borne inférieure des dates de début d'exécution.

Exemple

Graphe UET-UCT



Date de début d'exécution



Graphe critique

0 1 2 3 4 5

a	e	i	n	
b		j		
a	e	k		
b				
a	f			
b				
c	g			
b				
d	h			
c				
			l	o
			m	

Ordonnancement associé

Résultats positifs (3)

Modèle classique

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = 1$
- Nombre M de processeurs borné

algorithme $\frac{7}{3} - \frac{4}{3M}$-approché [Munier,Hanen 95]

Modèle hiérarchique

- $\forall i \in V, p_i = 1, \forall (i,j) \in E, c_{ij} = 1$ et $\epsilon_{ij} = 0$
- Nombre M de modules borné, m processeurs identiques par module

algorithme $(1 + (2 - \frac{1}{m})\rho^{\infty})$-approché [Bampis,Giroudeau,König 00]

Description

Ordonnancement-évolué

Si $|Y_i| \leq M$, alors nous affectons les tâches en une unité de temps tout en respectant les contraintes de précédence et les délais de communications.

Si $|Y_i| > M$, alors nous choisissons M tâches dans M modules différents et pour les tâches restantes nous appliquons un algorithme glouton d'affectation sur les processeurs. Nous exécutons après celles-ci les M tâches sélectionnées.

Exemple

	i	$i+1$	$i+2$
Π^9			
	a_i		
Π^8		$\boldsymbol{a_{i+1}}$	
	b_i	b_{i+1}	
Π^7	c_i	c_{i+1}	
	d_i	d_{i+1}	
Π^6		e_{i+1}	
	e_i	$\boldsymbol{f_{i+1}}$	
Π^5		g_{i+1}	
	$\boldsymbol{f_i}$	h_{i+1}	
Π^4		i_{i+1}	
	g_i		
Π^3	$\boldsymbol{h_i}$	$\boldsymbol{j_{i+1}}$	
	i_i		
Π^2	j_i	k_{i+1}	
	k_i		
Π^1	$\boldsymbol{l_i}$	l_{i+1}	
	m_i	m_{i+1}	

t	$t+1$	$t+2$	$t+3$	$t+4$	$t+5$	$t+6$	
a_i	i_i		g_{i+1}	d_{i+1}			Π^3_M
b_i	j_i	$\boldsymbol{f_i}$	h_{i+1}	e_{i+1}	$\boldsymbol{a_{i+1}}$		
c_i	k_i		b_{i+1}	k_{i+1}			Π^2_M
d_i	m_i	$\boldsymbol{h_i}$	c_{i+1}	i_{i+1}	$\boldsymbol{f_{i+1}}$		
e_i			l_{i+1}				Π^1_M
g_i		$\boldsymbol{l_i}$	m_{i+1}		$\boldsymbol{j_{i+1}}$		

Liste restreinte des contraintes de précédence

$$f_i \to f_{i+1};\quad h_i \to j_{i+1};\quad l_i \to l_{i+1}$$
$$f_i \to h_{i+1};\quad l_i \to m_{i+1};$$

Performance relative

-x le % d'instants dans σ^{∞} où $|Y_i| > M$

-$(1-x)$ le % d'instants dans σ^{∞} où $|Y_i| \leq M$

$$
\begin{aligned}
C_{max}^{M,h^*} &\leq (1-x)C_{max}^{h} + x\left(C_{max}^{h} + \sum_{i=0}^{C_{max}^{h}-1} \left\lceil \frac{|X_i|-M}{mM} \right\rceil\right) \\
&\leq (1 + x - \frac{x}{m})C_{max}^{h} + xC_{max}^{opt,M}
\end{aligned}
$$

Le pire-cas nous est donné quand $x = 1$, ainsi nous obtenons :

$$\rho^{M} \leq (2 - \frac{1}{m})\rho^{\infty} + 1 \Rightarrow \rho^{M} \leq \frac{17}{5} \text{ pour } \rho^{\infty} = \frac{8}{5}$$

Conclusions et Perspectives

- *Augmentation de la difficultée d'approximer et baisse de la qualité des algorithmes d'aproximation.*

- *Prendre en compte la spécificité des processeurs (vitesse,..),*
- *Adapter des heuristiques pour le problème classique d'ordonnancement avec communications dans le cas du modèle avec des communications hiérarchiques et mesurer leurs performances,*
- *Évaluer la différence de qualité de la solution pour un graphe donné entre le modèle homogène et le modèle hiérarchique.*

Définition

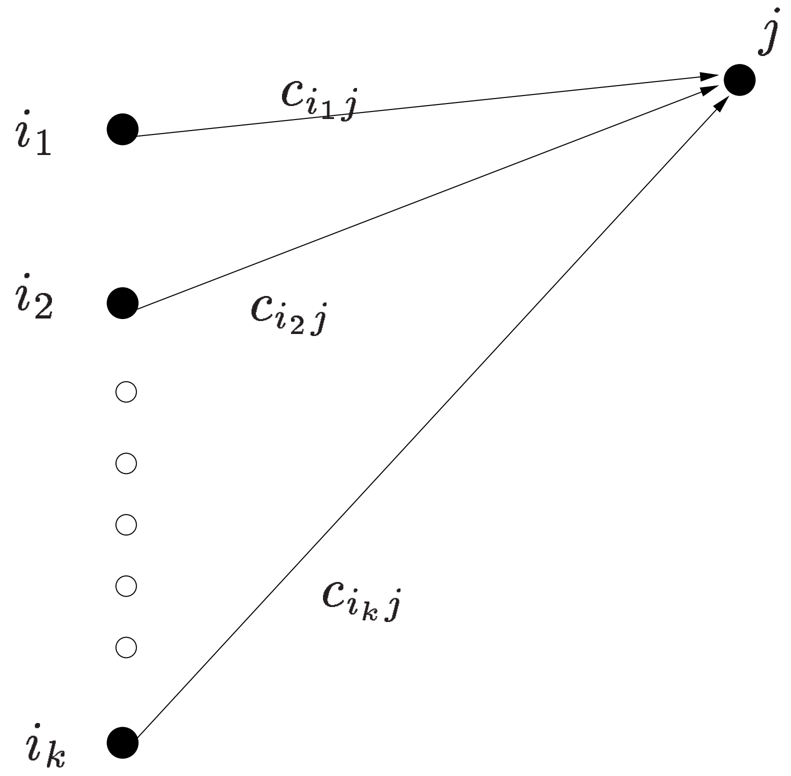


Tableau récapitulatif

Modèle	Modèle homogène		Modèle hiérarchique	
	Type de machine	Borne(s)	Type de machine	Borne(s)
UET-UCT	$\bar{P}$	$\frac{7}{6} \leq \rho \leq \frac{4}{3}$,	$\bar{P}(P2)$	$\frac{5}{4} \leq \rho \leq \frac{8}{5}$
SCT	$\bar{P}$	$\rho \leq \frac{2(1+\Phi)}{2\Phi+1}$,	$\bar{P}(P2)$	$\rho \leq \frac{12(1+\Phi)}{12\Phi+1}$
UET-UCT,biparti	P	$\frac{5}{4} \leq \rho$	$P2(P)$	$\frac{4}{3} \leq \rho$
UET-UCT	P	$\rho \leq \frac{7}{3} - \frac{4}{3m}$	$P(Pm)$	$\rho \leq (3 - \frac{1}{Mm})$
UET-UCT,dup	$\bar{P}$	$\rho = 1$	$\bar{P}(P2)$	$\rho = 1$