

# 7ème Conférence Internationale de Modélisation et Simulation

31 mars – 2 avril 2008 – Paris, France.

---

## Une Approche Coopérative pour l'Ordonnancement sous Incertitudes

**C. Briand<sup>1</sup> S. Ourari<sup>1,2</sup> et B. Bouzouia<sup>2</sup>**  
([sourari@laas.fr](mailto:sourari@laas.fr))

- (1) *Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS*
- (2) *Centre de Développement des Technologies Avancées*

# Introduction

## Problème Ordonnancement

### Statique et déterministe

- ❑ L'environnement d'application est incertain
  - **Perturbations** : Arrivée aléatoire des jobs et les paramètres de L'ordonnancement varient dans le temps;
  - **Conséquence** : Un ordonnancement de bonne qualité peut rapidement s'avérer mauvais, voire infaisable, suite à une perturbation

### → Ordonnancement robuste

### Organisation globale et centralisée

- ❑ Ressources partitionnées entre des acteurs possédant leur propre autonomie décisionnelle
  - Connaissance restreinte des paramètres de l'environnement
  - Objectives propres à chaque acteur,
  - Objectif global.

### → Ordonnancement par négociations entre acteurs

# Contexte

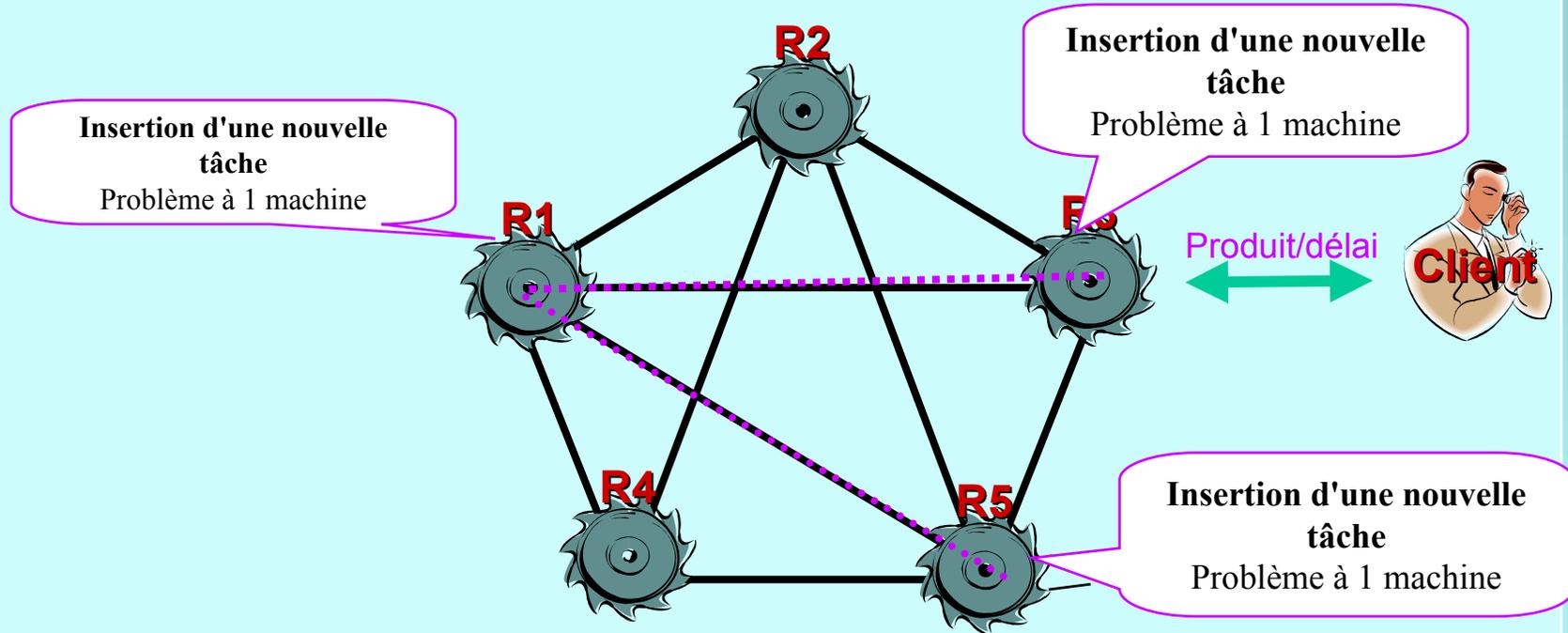
## ■ Intérêt de l'ordonnancement par coopération entre acteurs:

- Ordonnancement local robuste
  - *Résister aux perturbations locales et aux incertitudes liés à l'environnement*
  - *Caractérisation sur chaque ressource d'une famille flexibles d'ordonnancement par utilisation de condition de dominance.*
  - *Autonomie décisionnelle*
- Ordonnancement global par négociation entre les acteurs
  - *Les décisions d'ordonnancement sont distribuées à travers un réseau d'acteurs qui collaborent pour ne performance globale du système*
- Objectif : Compromis satisfaisant les exigences locale et globale.

## ■ Problème avec ressources disjonctives (job shop)

# Positionnement du problème

- Le problème **job shop** à plusieurs machines, noté  $J_n || C_{max}$



- Problème décomposé en  $m$  sous problèmes à une machine inter-dépendants, avec comme objectif, minimiser le retard algébrique (problème noté  $1 | r_i | L_{max}$ ).

- Introduction-Positionnement du problème
- Une approche d'ordonnancement robuste à une ressource
- Une approche pour l'ordonnancement coopératif
- Conclusion et Perspectives

# Une approche d'ordonnancement robuste à une ressource

## ■ Ordonnancement robuste

- Prise en compte des incertitudes pro-activement lors de la construction hors ligne des ordonnancements
- Produire une solution / une famille de solutions flexibles dont la qualité est garantie a priori vis-à-vis d'un ensemble de perturbations
- Utilisation des notions de :
  - *Flexibilité temporelle*
  - *Flexibilité séquentielle*
    - Concept de groupe d'opération permutable
    - Condition de dominance: **Théorème des pyramides**

# Utilisation de conditions de dominance

## ■ Un théorème de dominance (Erschler et al., 1983)

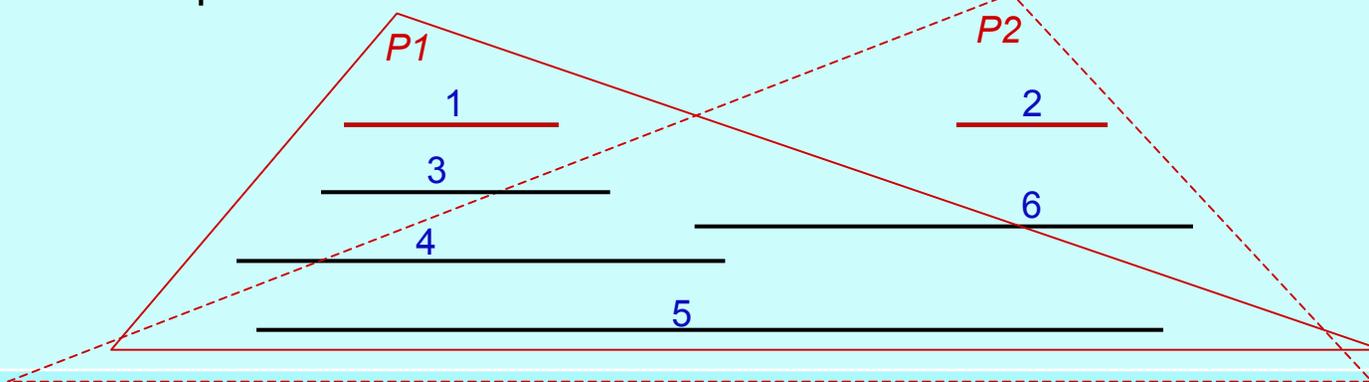
### – Fondements

- *Problème 1 machine avec  $r_i$  (date de disponibilité) et  $d_i$  (échéance)*
- *Corps d'hypothèses : Ordre relatif des  $r_i$  et  $d_i$ , durées quelconques*
- *Un intervalle pour chaque travail :  $i \rightarrow [r_i, d_i]$*
- *Un problème  $\rightarrow$  Une structure d'intervalles*
- *Fondée sur les notions de sommet et de S-pyramide*
- *Dominance vis-à-vis de  $T_{max}$ ,  $L_{max}$  et de l'admissibilité*

### – Exemple

CH Restreint :

$$r_4 < r_5 < r_3 < r_1 < d_1 < d_3 < r_6 < d_4 < r_2 < d_2 < d_5 < d_6$$



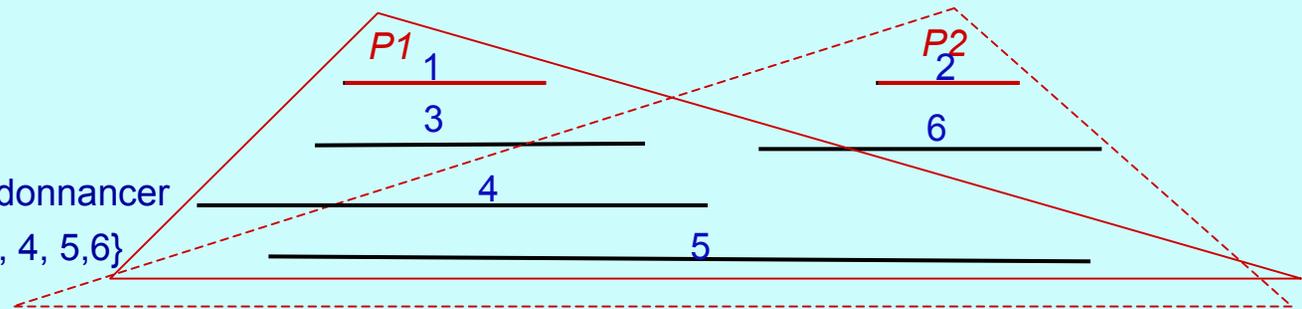
# Théorème des pyramides

## Le nombre de séquences caractérisées est connu

$$\text{Card}(S) = \prod_{q=1}^{q=N} (q+1)^{n_q} \quad n_q \text{ est le nombre de travaux appartenant exactement à } q \text{ pyramides}$$

## Exemple

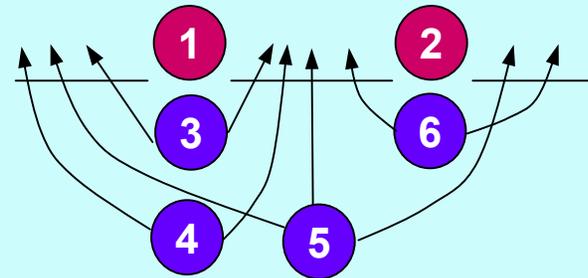
- 6 tâches à ordonnancer  
 $P = \{1, 2, 3, 4, 5, 6\}$



- Nombre de pyramides :  $N = 2$

- $q = 1 : n_1 = 3$  (travail 3, 4 et 6)
- $q = 2 : n_2 = 1$  (travail 5)

→  $\text{Card}(S) = (1+1)^3(2+1)^1 = 24$

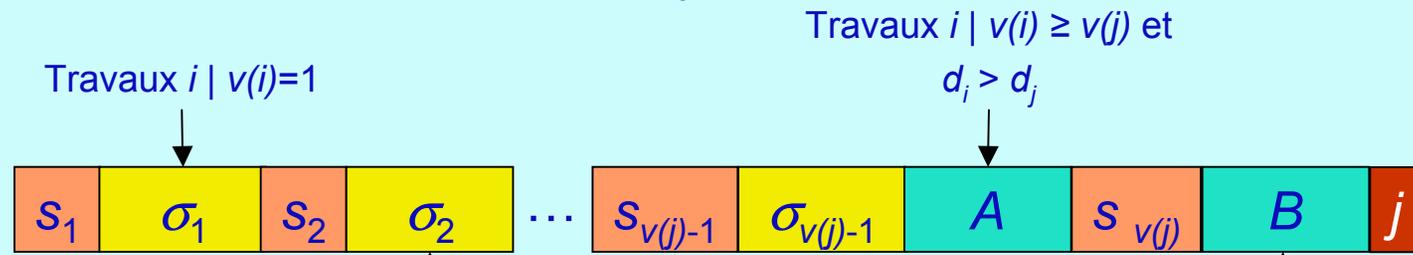


## 24 séquences dominantes (parmi les $6! = 720$ seq. possibles)

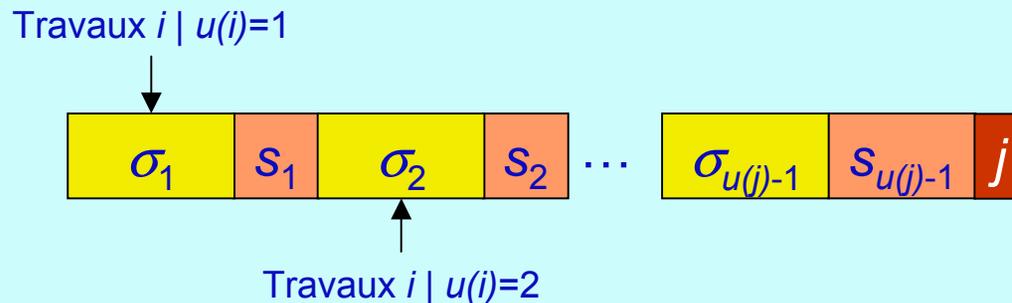
4 < 5 < 3 < 1 < 6 < 2	4 < 3 < 1 < 6 < 2 < 5	5 < 3 < 1 < 4 < 6 < 2	3 < 1 < 4 < 6 < 2 < 5
4 < 5 < 3 < 1 < 2 < 6	4 < 3 < 1 < 2 < 5 < 6	5 < 3 < 1 < 4 < 2 < 6	3 < 1 < 4 < 2 < 5 < 6
4 < 5 < 1 < 3 < 6 < 2	4 < 1 < 3 < 5 < 6 < 2	5 < 1 < 3 < 4 < 6 < 2	1 < 3 < 4 < 5 < 6 < 2
4 < 5 < 1 < 3 < 2 < 6	4 < 1 < 3 < 5 < 2 < 6	5 < 1 < 3 < 4 < 2 < 6	1 < 3 < 4 < 5 < 2 < 6
4 < 3 < 1 < 5 < 6 < 2	4 < 1 < 3 < 6 < 2 < 5	3 < 1 < 4 < 5 < 6 < 2	1 < 3 < 4 < 6 < 2 < 5
4 < 3 < 1 < 5 < 2 < 6	4 < 1 < 3 < 2 < 5 < 6	3 < 1 < 4 < 5 < 2 < 6	1 < 3 < 4 < 2 < 5 < 6

# Évaluation de la performance d'un ensemble dominant

- Séquence la plus défavorable pour  $j$ :  $L_j^{\min}$



- Séquence la plus favorable pour  $j$ :  $L_j^{\max}$



- Complexité temporelle :  $O(n \cdot \log(n))$

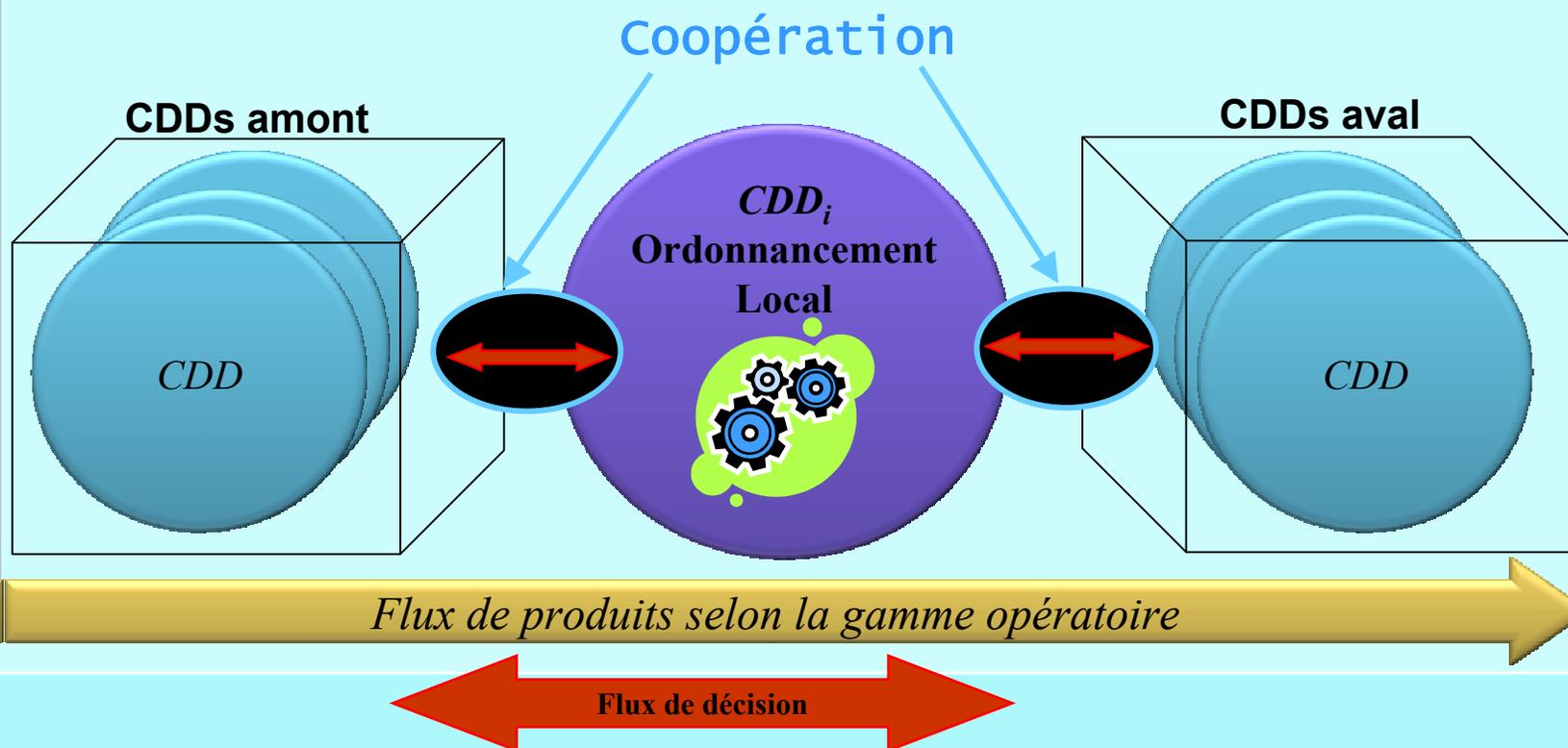
- Dédution de  $[s_j^{\min}, s_j^{\max}]$  et  $[s_j^{\min}, s_j^{\max}]$  sachant  $\max_{i \in V} (L_i^{\min}) \leq L_{\max} \leq \max_{i \in V} (L_i^{\max})$

- Introduction
- Positionnement du problème
- Une approche d'ordonnancement robuste à une ressource
- Une approche pour l'ordonnancement coopératif
- Conclusion et Perspectives

# Contexte

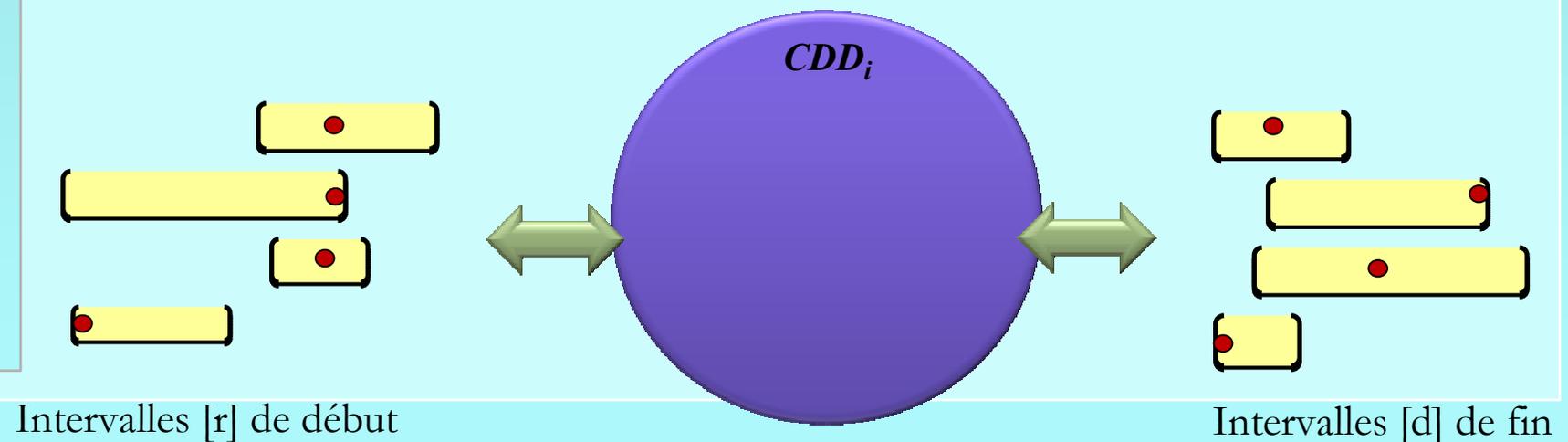
## ■ Ordonnancement coopératif

- $m$  centres de décisions, chacun gérant son propre ordonnancement local
- Chaque centre possède sa propre autonomie décisionnelle
- Coopération point à point entre les CDDs voisins => Cohérence des délais
- Coopération décomposée: Négociation / Renégociation / Coordination



# Hypothèses

- 1 CDD  $\Leftrightarrow$  1 machine
  - Ordonnancement local flexible à une machine
- Ensemble CDD liés par les gammes opératoires des jobs  $\Leftrightarrow$  jobshop
  - Ordonnancement global distribué par coopération entre CDDs
- Objectif coopération = Négocier les dates de livraison/consommation
  - Flexibilité sur les délais  $\Leftrightarrow$  autonomie décisionnelle
  - Intervalles de début  $[r]$  et de fin  $[d]$
- Cohérence (ordonnancement local non flexible)
  - L'ordonnancement local doit définir des dates de début et de fin de tâches compatibles avec les intervalles  $[r]$  et  $[d]$

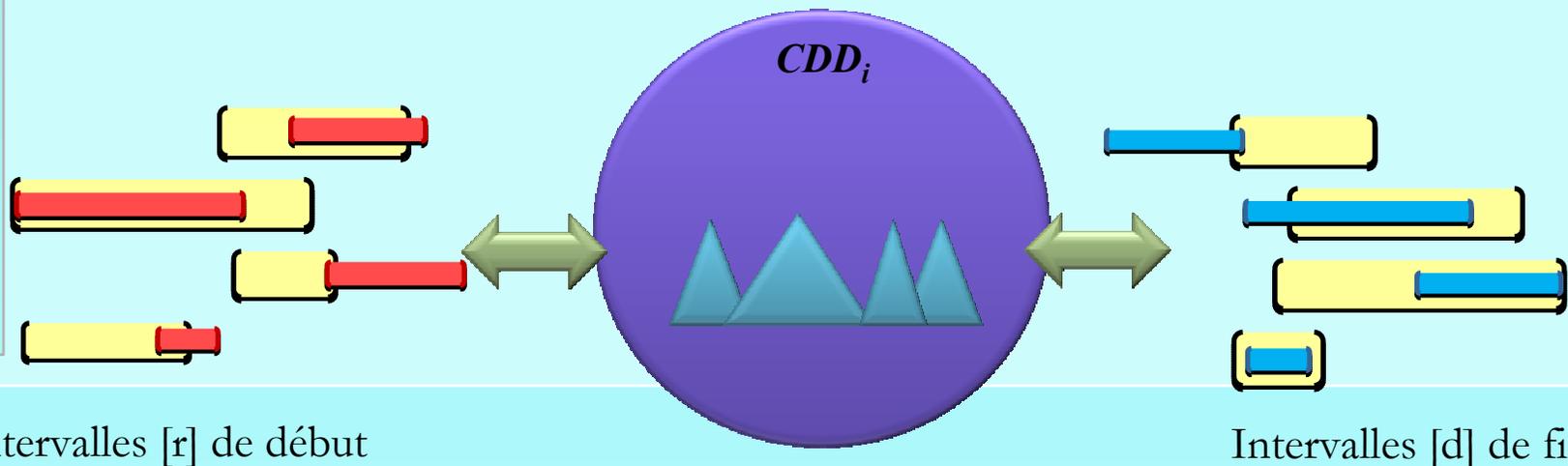


# Processus de coopération inter-machines

## ■ Ordonnements locaux robustes

- Flexibilité séquentielle :
  - *ordre partiel sur les tâches : théorème des pyramides*
  - *Caractérisation d'un ensemble de séquences dominantes :*
  - *→ Dates au mieux et au pire de début et de fin pour chaque tâche (intervalles [s] et [f])*
- Cohérence locale=>Engagement
  - *Consommation au plus tôt dans l'intervalle de disponibilité [r]*
  - *Livraison au plus tard dans l'intervalle [d]*

$$r_{uv}^{\min} \leq s_{uv}^{\min} \leq r_{uv}^{\max} \quad \text{et} \quad d_{uv}^{\min} \leq f_{uv}^{\max} \leq d_{uv}^{\max}$$



# Processus de coopération inter-machines

## ■ Ordonnancement coopératif

- Le plan d'exécution global résulte de la concaténation d'ordonnements locaux, élaborés par:
  - Négociation
    - Initiée lorsqu'un CDD prend en charge une nouvelle commande
    - Constituée d'une suite de proposition et contre proposition
    - Aboutissant à la définition d'un intervalle de consommation/livraison.
  - Renégociation  $\approx$  conversation
    - Initiée lorsqu'un CDD souhaite modifier un accord contracté
  - Coordination
    - Echange d'information pour vérification de la cohérence des décisions

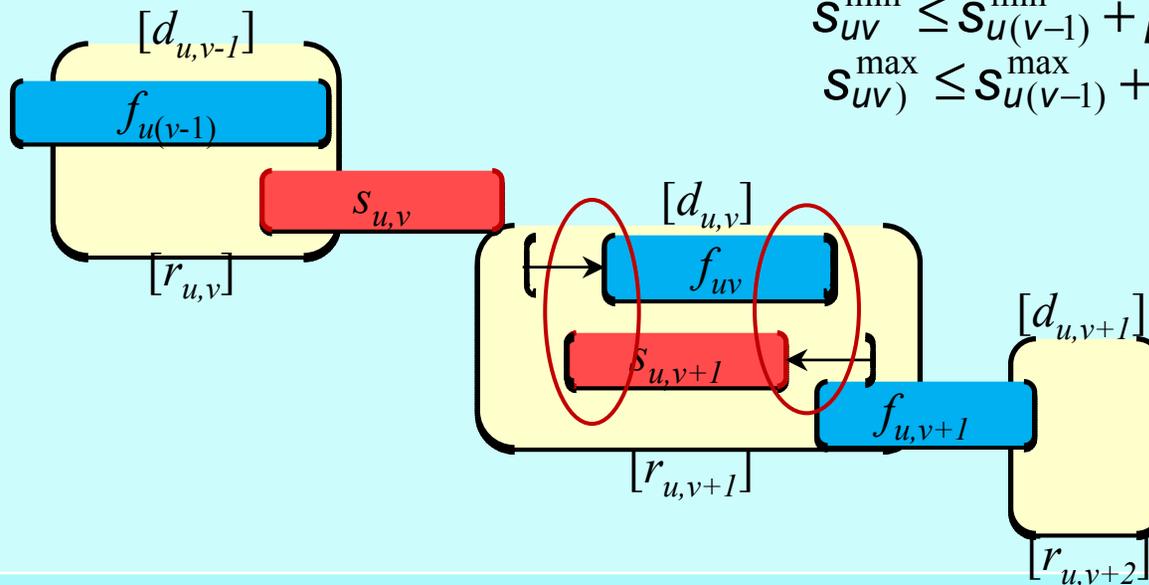
# Processus de coopération inter-machines

## Coordination entre CDD

- Les CDDs se communiquent les intervalles [s] et [f] des tâches qu'ils gèrent au fur et à mesure qu'ils évoluent
- Maintenir la cohérence globale des intervalles [s] et [f]
- Cette coordination peut induire un re-ordonnancement local entre les dates de début au mieux et au pire:

### Contraintes de cohérence globale

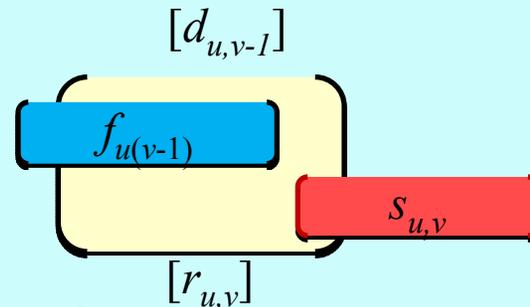
$$s_{uv}^{\min} \leq s_{u(v-1)}^{\min} + p_{u(v-1)} = f_{u(v-1)}^{\min}$$
$$s_{uv}^{\max} \leq s_{u(v-1)}^{\max} + p_{u(v+1)} = f_{u(v+1)}^{\max}$$



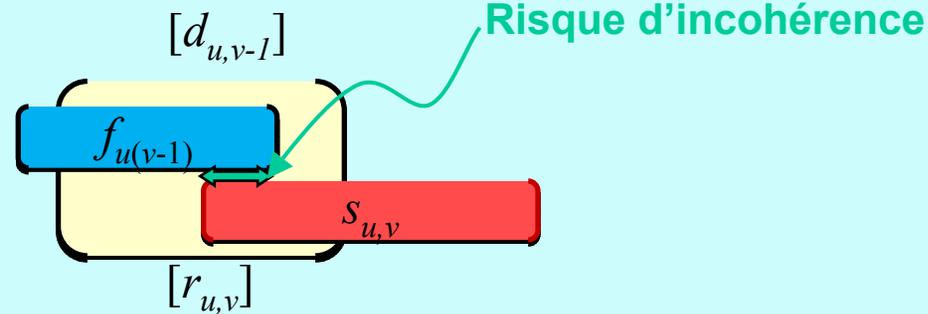
# Notion de risque d'incohérence

## ■ Cohérence des intervalles $[r]$ et $[d]$

- Avec risque nul



- Avec risque non nul

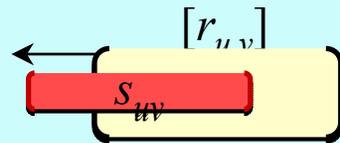


- Le risque d'incohérence est d'autant plus grand que l'intervalle de chevauchement entre  $f_{u(v-1)}$  et  $s_{uv}$  est important

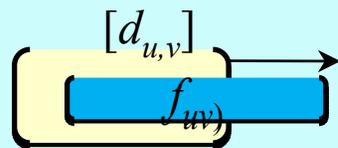
# Renégociation

## ■ Renégociation

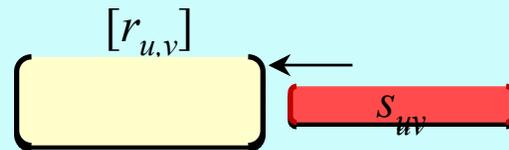
- Suite à une détection d'incohérence ← re-ordonnancement / aléas



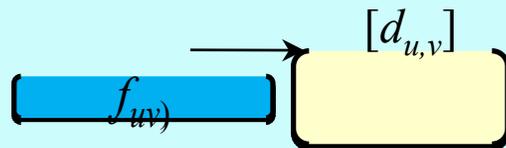
On veut pouvoir commencer ( $w$ ) plus tôt



On veut pouvoir finir ( $w$ ) plus tard



Sur-autonomie : ( $u(v-1)$ ) peut être finie plus tard



Sur-autonomie : ( $u(v+1)$ ) peut commencer plus tôt

# Négociation

- **Une négociation a lieu quand un nouveau job  $u$  caractérisé par une date d'échéance  $[d_u]$  arrive**
  - Toute les tâches associées au nouveau job  $u$  sont affectées aux ressources correspondantes
- **Il s'agit de déterminer les intervalles :**
  - de début  $[s_{uj}]$  et de fin  $[f_{uj}]$  au mieux et au pire pour chaque tâche de ce job de manière à respecter les contraintes de cohérence internes
  - de début  $[s]$  et de fin  $[f]$  aux mieux et aux pires des autres tâches si l'insertion du nouveaux job crée des incohérences
- **On suppose que les intervalles existants avant l'arrivée du job  $u$  sont cohérents**

# Négociation/Renégociation

## ■ Objectifs

- L'insertion d'une nouvelle tâche dans un ordonnancement local induit la modification des intervalles  $[s]$  et  $[f]$  des tâches existantes
- Il faut insérer la tâche de sorte à :
  - Respecter les différentes contraintes
  - minimiser le risque d'incohérence
  - Il faut cependant conserver de la flexibilité → résister aux aléas externes ou internes
- Nécessité d'un compromis

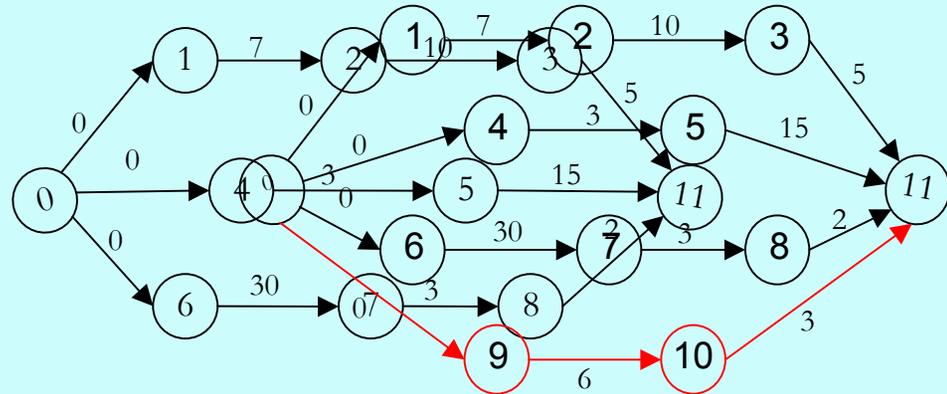
## ■ Problèmes à résoudre:

- Déterminer un ordre total entre  $r$  et  $d$  compatibles avec  $[r]$  et  $[d]$
- Déterminer les valeurs de  $r$  et  $d$  pour déduire les  $[s]$  et  $[f]$

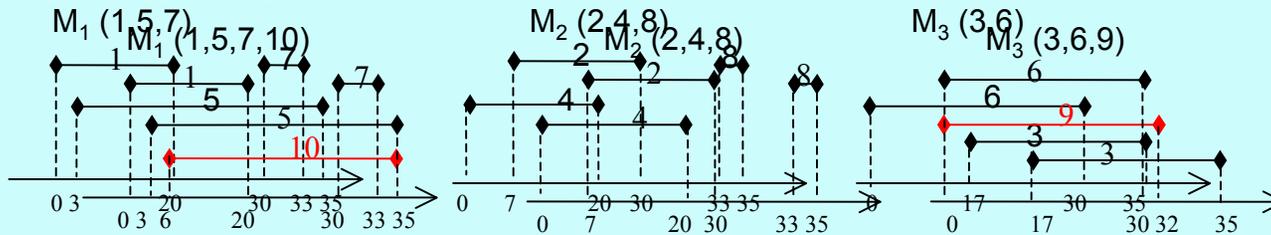
# Exemple

- Délais réduits à un point est hédon pour les autres jobs par propagation de contraintes

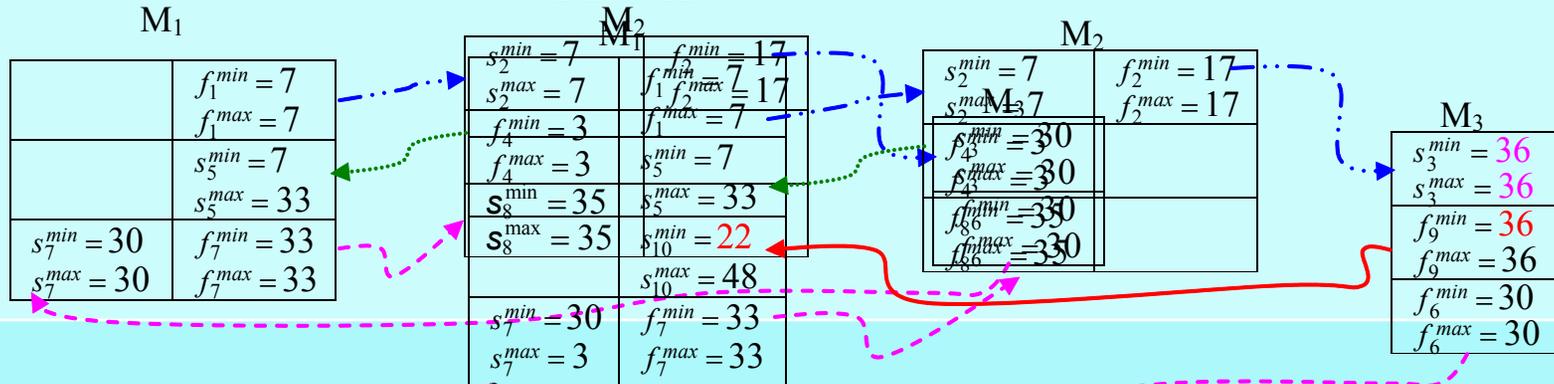
travail	travail tâche	tâche $p_i$	$p_i$ $M_j$	$M_j$
1	1	7	110	2
1	2	10	25	3
1	3	5	33	2
2	4	35	215	1
2	5	15	130	3
3	6	30	33	1
3	7	38	12	2
3	8	29	26	3
	4	10	3	1



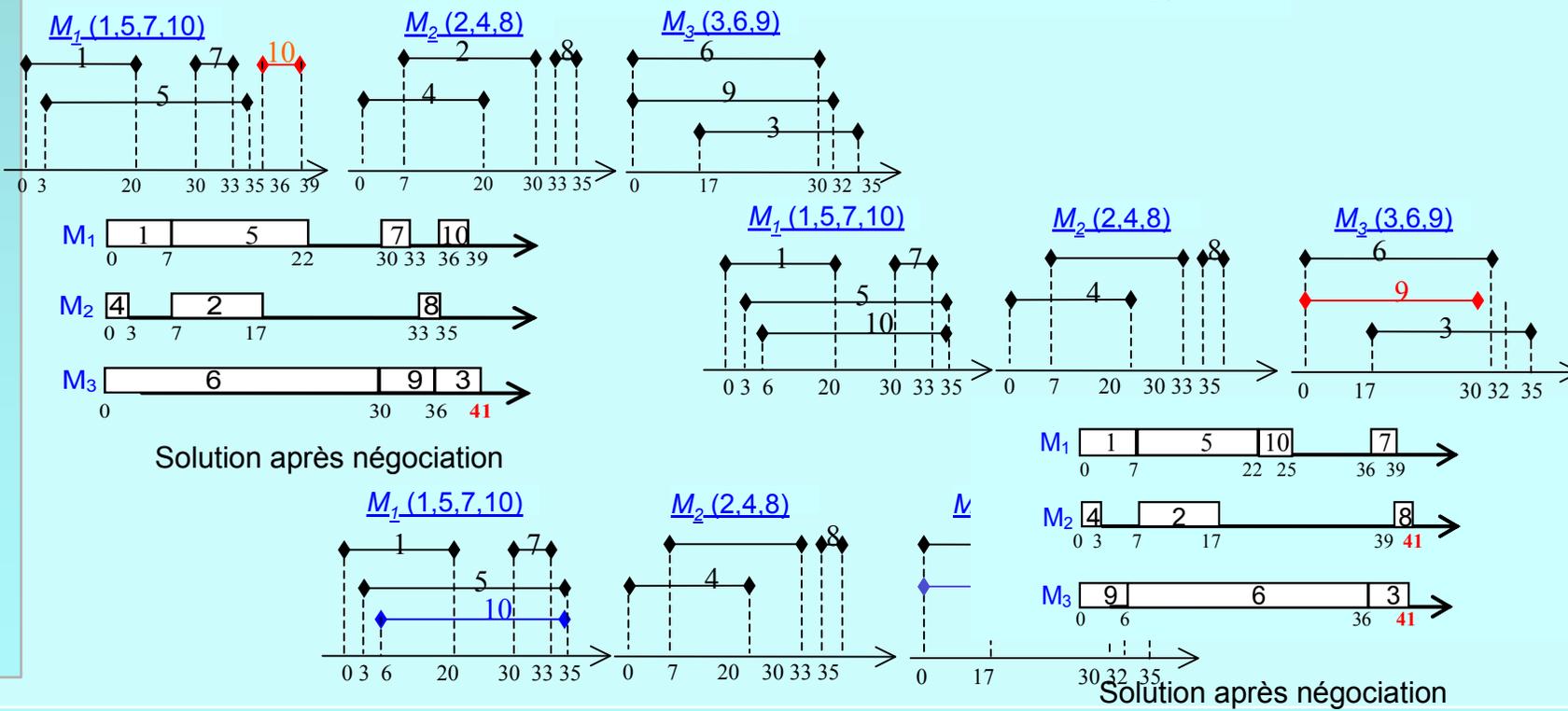
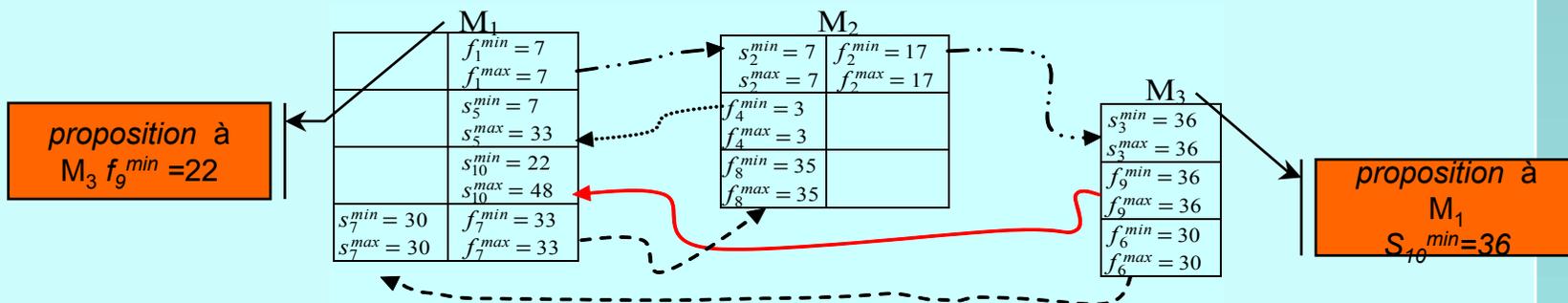
- Structures d'intervalles



- Cohérences des intervalles [s] et [f]



# Structure décisionnelle distribuée



(Solution optimale donnée dans [1])

# Conclusion

- Une approche distribuée basée sur la coopération inter-ressources.
  - Chaque ressource gère un ordonnancement local robuste,
  - Décisions d'ordonnancement résultent de la négociation/renégociation des ressources
- Comment organiser la coopération (quel type de relation) ?
  - Contraintes de cohérence, risque d'incohérence et taux de flexibilité
  - Intervalle de consommation/livraison
- Comment formaliser la coopération (quand, comment et sur quoi coopérer) ?
  - Coopération initiée à l'occurrence d'un nouveau job, ou d'un aléa
  - Mode de coopération point à point
  - Coopération = négociation, coordination, renégociation
  - flexibilité sur les délais
- **Perspectives :**
  - Définir une heuristique de résolution
  - Implémenter l'approche en vue de sa validation et de son amélioration.

*Merci de votre attention!*

?