9ème congrès des doctorants de l'Ecole Doctorale Systèmes -

Institut National des Sciences Appliquées de Toulouse - 22 mai 2008

Utilisation de Conditions de Dominance dans une Perspective d'Ordonnancement Coopératif

S. Ourari *(sourari@laas.fr)*

Directeur de thèse: C. Briand *(briand@laas.fr)*

Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS
7, Av. du Colonel Roche
31077 Toulouse Cedex 4

Introduction

- **Ordonnancement**
  - Organiser dans le temps la réalisation de tâches interdépendantes, compte tenu de contraintes de temps et de ressources
  - Les paramètres du problème sont supposés connues et déterministes, et la fonction d'ordonnancement est globale

- **Or 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 : définir une organisation capable de résister aux perturbations**

- **Or les ressources sont souvent partitionnées entre des acteurs possédant leur propre autonomie décisionnelle**
  - Ex : Gestion de projet, Chaîne logistique, Grille de calcul, …
  - Chaque acteur a une connaissance restreinte de son environnement
  - Objectives locaux vs. objectif global.

  ➔ **Ordonnancement coopératif : l'ordonnancement global résulte d'une négociation entre acteurs**

Introduction

- **Hypothèses : une approche robuste ET coopérative**

  - Ordonnancement local robuste: Problème à une machine $1|r_i|L_{max}$

    - *Résister non seulement aux perturbations locales mais surtout aux incertitudes liées à une mauvaise connaissance de l'environnement*

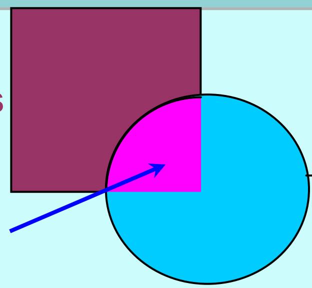
    - ***Méthode** : caractérisation sur chaque acteur d'une famille flexibles d'ordonnancement de performances au mieux et au pire connues.*
      - Utilisation des notions de flexibilité séquentielle
        - **Condition de dominance:** Théorème des pyramides

  - Ordonnancement global obtenu par négociations entre les acteurs
    - *Les décisions d'ordonnancement sont distribuées à travers un réseau d'acteurs qui collaborent en vue d'assurer une performance globale satisfaisante*

  - Objectif : Compromis performance globale / flexibilité locale

Utilisation de conditions de dominance

- **Principes**



- **Avantages**
  - Dominance => caractérisation de solutions "plus admissibles" ou "plus optimales" que les autres
  - CH restreint => Les paramètres du problème ne sont pas connus précisément, insensibilité aux variations de paramètres

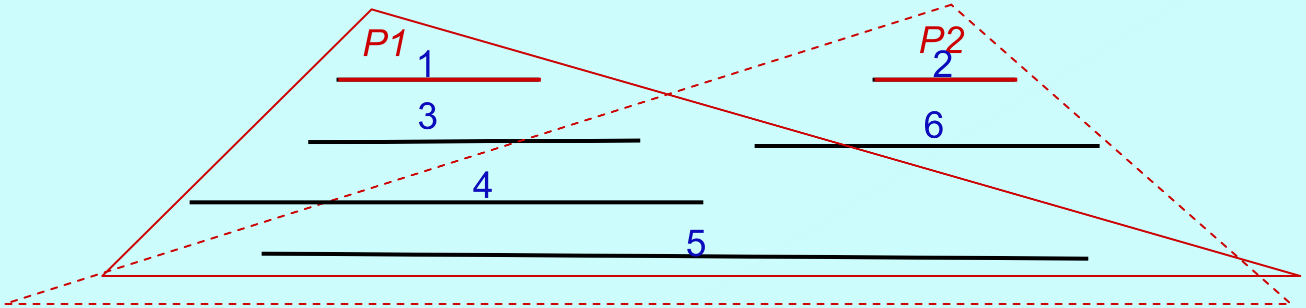
- **Un théorème de dominance (*Erschler et al., 1983*)**
  - Fondements
    - *Problème 1 machine avec* r_i *(date de disponibilité) et* d_i *(échéance)*
    - *Corps d'hypothèses :* *<u>Ordre relatif</u> des* r_i *et* d_i, *<u>durées quelconques</u>*
    - *Un intervalle pour chaque travail :* $i \rightarrow [r_i, d_i]$
    - *Un problème* ➔ *Une structure d'intervalles*
    - *Fondée sur les notions de sommet et de S-pyramide*
    - *Définit un ensemble de séquences dominantes*
    - *Dominance vis-à-vis l'admissibilité et de* T_{max} , L_{max}

Théorème des pyramides

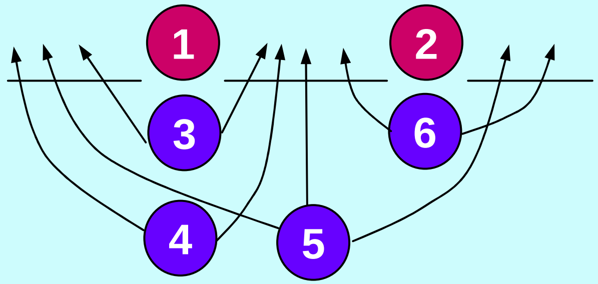
CH Restreint :

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

- Exemple



- Cardinalité de l'ensemble dominant



$$Card\ (S) = \prod_{q=1}^{q=N} (q+1)^{n_q}$$

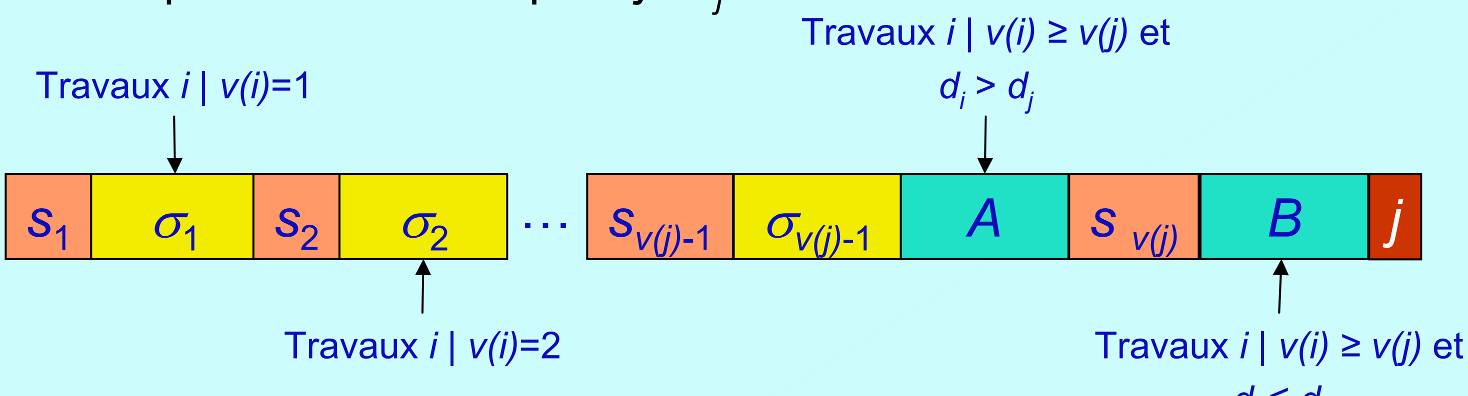
n_q est le nombre de travaux appartenant exactement à q pyramides

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

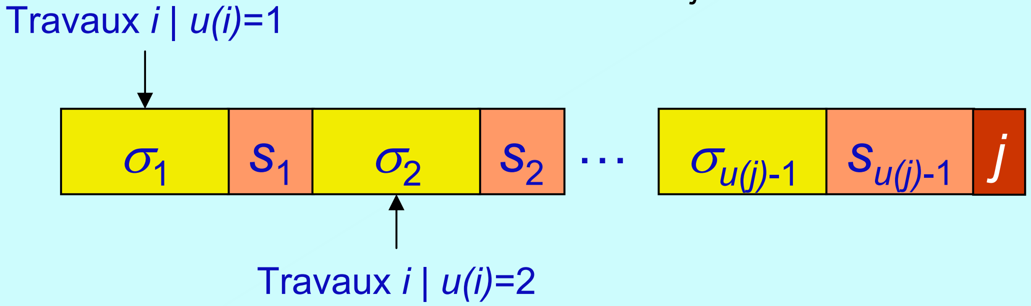
$4 \prec 5 \prec 3 \prec 1 \prec 6 \prec 2$	$4 \prec 3 \prec 1 \prec 6 \prec 2 \prec 5$	$5 \prec 3 \prec 1 \prec 4 \prec 6 \prec 2$	$3 \prec 1 \prec 4 \prec 6 \prec 2 \prec 5$
$4 \prec 5 \prec 3 \prec 1 \prec 2 \prec 6$	$4 \prec 3 \prec 1 \prec 2 \prec 5 \prec 6$	$5 \prec 3 \prec 1 \prec 4 \prec 2 \prec 6$	$3 \prec 1 \prec 4 \prec 2 \prec 5 \prec 6$
$4 \prec 5 \prec 1 \prec 3 \prec 6 \prec 2$	$4 \prec 1 \prec 3 \prec 5 \prec 6 \prec 2$	$5 \prec 1 \prec 3 \prec 4 \prec 6 \prec 2$	$1 \prec 3 \prec 4 \prec 5 \prec 6 \prec 2$
$4 \prec 5 \prec 1 \prec 3 \prec 2 \prec 6$	$4 \prec 1 \prec 3 \prec 5 \prec 2 \prec 6$	$5 \prec 1 \prec 3 \prec 4 \prec 2 \prec 6$	$1 \prec 3 \prec 4 \prec 5 \prec 2 \prec 6$
$4 \prec 3 \prec 1 \prec 5 \prec 6 \prec 2$	$4 \prec 1 \prec 3 \prec 6 \prec 2 \prec 5$	$3 \prec 1 \prec 4 \prec 5 \prec 6 \prec 2$	$1 \prec 3 \prec 4 \prec 6 \prec 2 \prec 5$
$4 \prec 3 \prec 1 \prec 5 \prec 2 \prec 6$	$4 \prec 1 \prec 3 \prec 2 \prec 5 \prec 6$	$3 \prec 1 \prec 4 \prec 5 \prec 2 \prec 6$	$1 \prec 3 \prec 4 \prec 2 \prec 5 \prec 6$

Évaluation de la performance d'un ensemble dominant

- Séquence la plus défavorable pour *j*: L_j^{max}



- Séquence la plus favorable pour *j*: L_j^{min}



- Complexité temporelle : $O(n.log(n))$

- Déduction de $[s_j^{min}, s_j^{max}]$ et $[f_j^{min}, f_j^{max}]$

Problèmes considérés

- Problème à une ressource

  - Généraliser la condition de dominance d'Erschler *et al.* au *problème de minimisation du nombre de travaux en retard.*

- Cas multi-ressources: Une approche coopérative

  - *Profitant de la flexibilité séquentielle engendrée par l'utilisation de l'ordre partiel dominant.*
    - *Chaque ressource gère son propre ordonnancement local pour faire face aux aléas (ordonnancement robuste).*
    - *L'ordonnancement global résulte alors d'une coopération entre les différentes ressources.*

Problème à une machine

- **Résultats**

1. Utilisation de la condition de dominance pour la minimisation du nombre de travaux en retard

   - *On montre sous quelles conditions, le théorème est dominant vis-à-vis de la minimisation du nombre de travaux en retard*

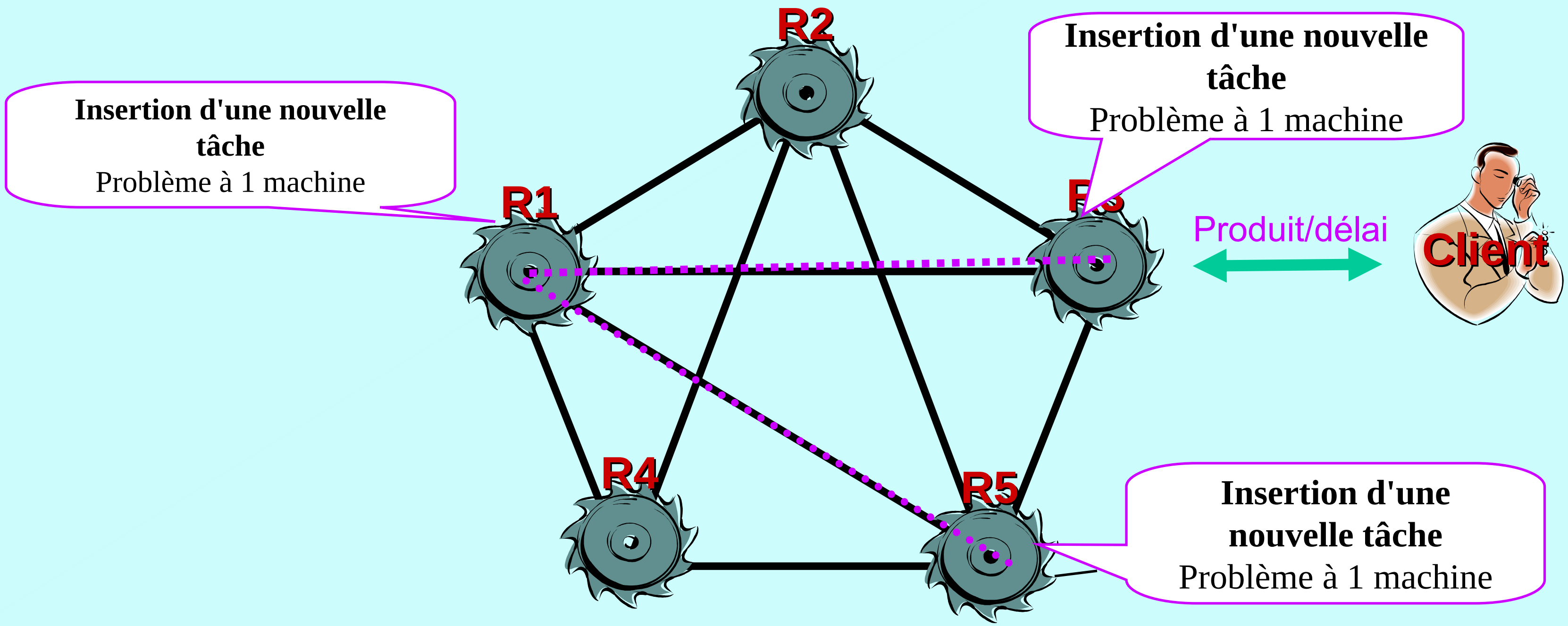
2. Modélisation du problème en un programme linéaire en nombre entiers

   - *Critère d'admissibilité*
   - *Critère de minimisation du nombre de travaux en retard*

- Introduction
- Problème à une ressource
- Problème multi-ressources: Une approche coopérative
- Conclusion et Perspectives

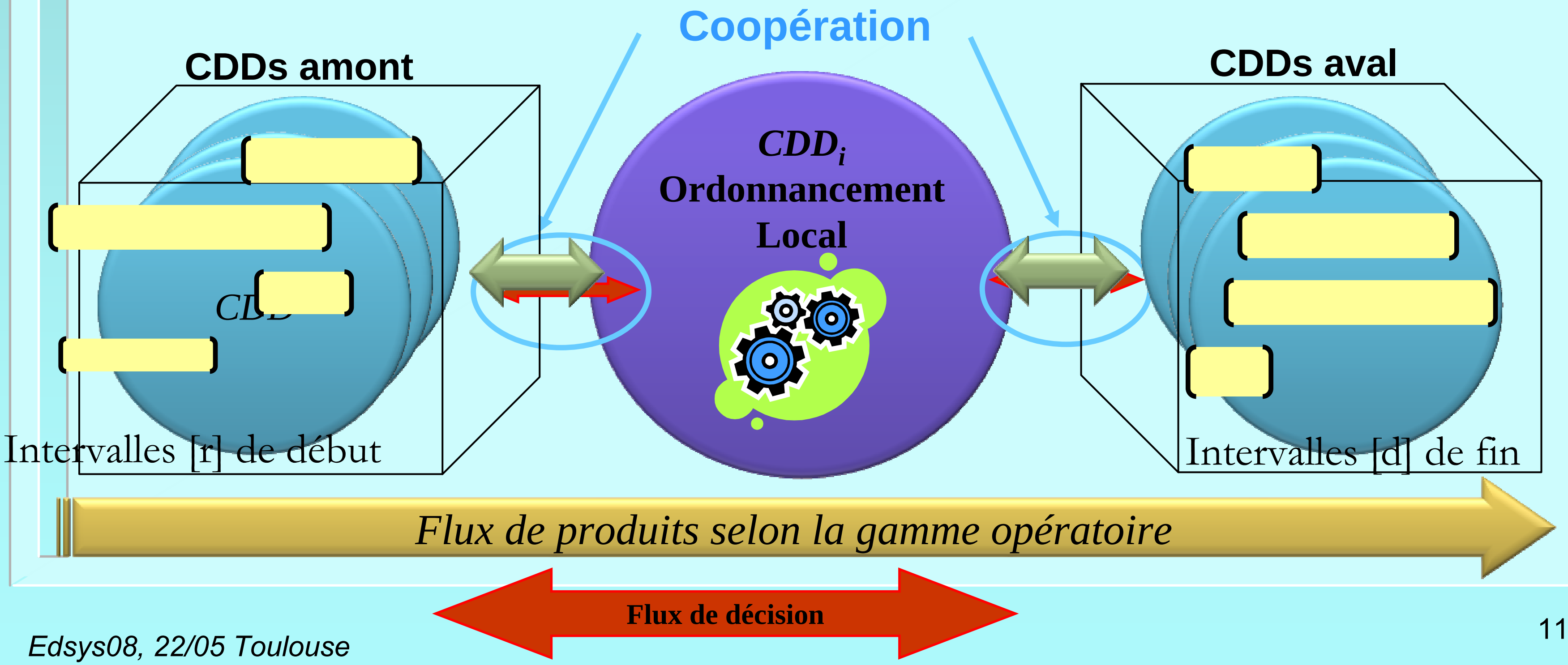
Introduction

- Le problème **job shop** à plusieurs machines, noté ***Jn||Cmax***
  - Les liens entre ressources dépendent des gammes opératoires
  - Problème décomposable en *m* sous problèmes à une machine inter-dépendants, avec comme objectif, minimiser le retard algébrique (problème noté *1|ri|Lmax*).



Contexte

- **Ordonnancement coopératif**
  - *m* centres de décisions, chacun gérant son propre ordonnancement local
  - Chaque centre possède sa propre autonomie décisionnelle
  - Lié aux autres centres par la gamme opératoire
  - Coopération point à point entre les CDDs voisins => Cohérence des délais
- Objectif coopération = Négocier les intervalles de livraison/consommation
  - Flexibilité sur les délais /Intervalles de début [*r*] et de fin [*d*]



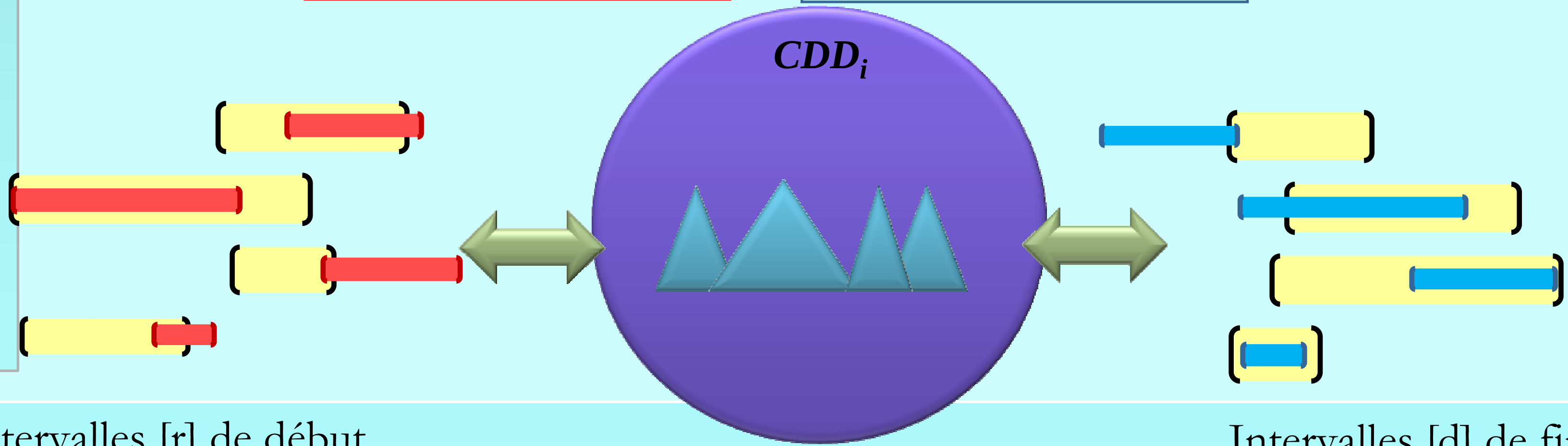
Situations de coopération

- **Objet de coopération ➔** intervalles de livraison/consommation
- **3 situations**
  - Négociation
    - Initiée lorsqu'un CDD prend en charge une nouvelle opération
    - Constituée d'une suite de propositions et de contre propositions
    - Aboutissant à la définition d'un intervalle de consommation/livraison.
  - Renégociation ≈ conversation
    - Initiée lorsqu'un CDD souhaite modifier un accord contracté
    - Constituée d'une suite de propositions et de contre propositions
    - Aboutissant à la modification d'un intervalle de consommation/livraison.
  - Coordination (par défaut)
    - Échange d'informations pour la synchronisation des CDDs
    - Les ordonnancements évoluent au cours du temps
      - Lorsque de nouveaux travaux apparaissent
      - Lorsque les opérations sont réalisées
      - Lorsque des perturbation surviennent

Contraintes de cohérence locale

- Ordonnancements 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])*
- Assurer la cohérence locale=> respect des engagements
    - *Consommation au plus tôt dans l'intervalle de disponibilité [r]*
    - *Livraison au plus tard dans l'intervalle [d]*

$$\boxed{r_{uv}^{\min} \le s_{uv}^{\min} \le r_{uv}^{\max}} \text{ et } \boxed{d_{uv}^{\min} \le f_{uv}^{\max} \le d_{uv}^{\max}}$$



Intervalles [r] de début

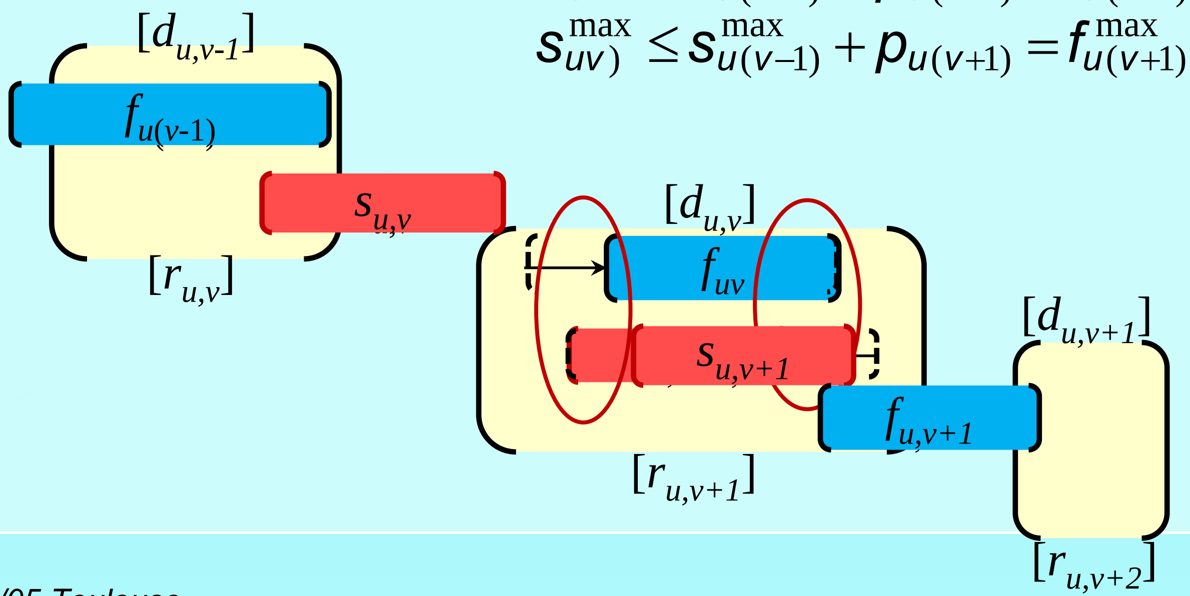
Intervalles [d] de fin

Contraintes de cohérence globales

- Coordination entre CDD
  - *Les CDDs se communiquent les intervalles [s] et [f] des opérations qu'ils gèrent au fur et à mesure que ces intervalles évoluent*
  - *Maintenir la cohérence globale des intervalles [s] et [f]*

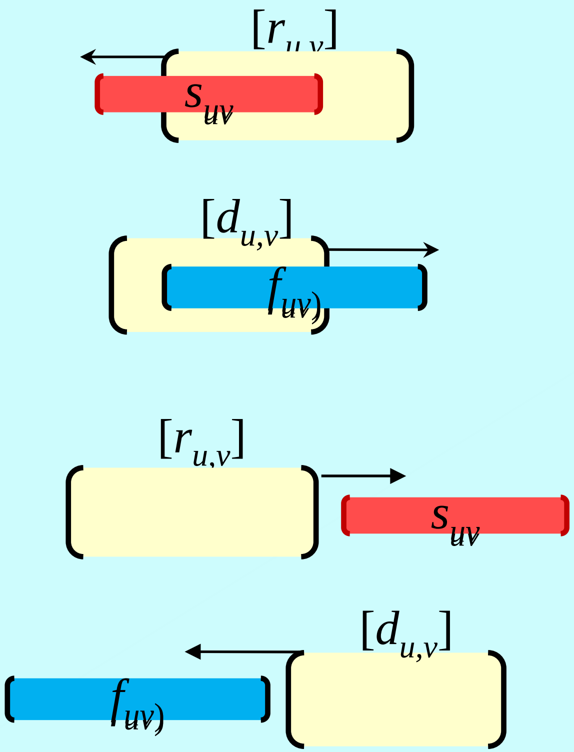
Contraintes de cohérence globale

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



Renégociation

- **Renégociation**
  - Suite à une détection d'incohérence locale 🡨 re-ordonnancement / aléas



On veut pouvoir commencer (*uv*) plus tôt

On veut pouvoir finir (*uv*) plus tard

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

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

Méthode de Négociation/Renégociation

- **Objectifs**
  - Déterminer ou modifier un intervalle de livraison/consommation
  - ➔induit la modification des intervalles [*s*] et [*f*] des tâches existantes
  - Il faut calculer l'intervalle de sorte à :
    - Respecter les différentes contraintes (locale et globale)
    - Il faut cependant <u>conserver de la flexibilité</u> ➔ résister aux aléas externes ou internes
  - Nécessité d'un <u>compromis</u>

- **Problèmes à résoudre:**
  - Déterminer un ordre total entre *r* et *d* compatibles avec les intervalles [r] et [d] contractés
  - Déterminer les valeurs de *r* et *d* pour déduire les [s] et [f] (application du théorème des pyramides)

Conclusion-Perspective

- **Problème à une machine**
  - Définir sous quelles conditions le théorème des pyramide est dominant vis-à-vis de la minimisation du nombre de travaux en retard.
  - Le problème a été modélisé sous forme d'un programme linéaire en nombre entiers

  Perspectives : Définir des bornes inférieures et supérieures

- **Problème à plusieurs ressources**
  - Une approche distribuée basée sur une coopération inter-ressources
    - *Chaque ressource gère un ordonnancement local robuste,*
    - *Les décisions d'ordonnancement sont négociées/renégociées entre ressources dynamiquement*
  - Comment organiser la coopération ?
    - *Intervalles de consommation/livraison contractés entre paire de ressources*
  - Comment formaliser la coopération (quand, comment et sur quoi coopérer) ?
    - *Coopération initiée à l'occurrence d'un nouveau job, ou d'un aléa*
    - *Coopération = négociation, coordination, renégociation*
    - *Flexibilité sur les délais*

  Perspectives : Mettre au point une heuristique pour la négociation des intervalles de livraison / consommation

Merci de votre attention!

Problème à une machine

Définition d'une séquence *maître-Pyramide: SMP*

- Pour éviter d 'énumérer toutes les séquences dominantes:

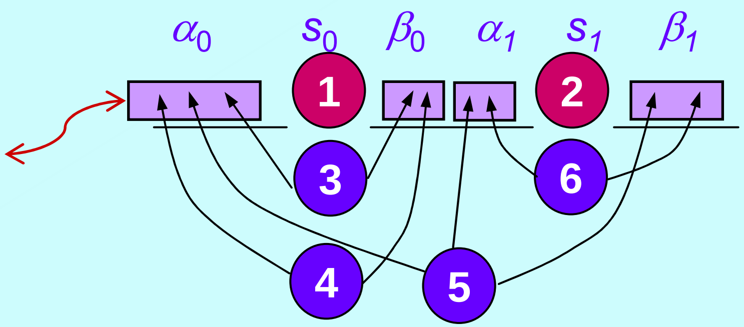
Séquence Maître-Pyramide

$$\boxed{\sigma_{\Delta} = (\alpha_0\, s_0\, \beta_0\, \alpha_1\, s_1\, \beta \ldots\, \alpha_i\, s_i\, \beta_i \ldots \alpha_m\, s_m\, \beta_m)}$$

  - m est le nombre total de sommets ;
  - α_i est l'ensemble des travaux appartenant à P_i et n'appartenant P_{i-1} classés $r_i\uparrow$
  - β_i est l'ensemble des travaux appartenant P_i classés d_i

- Exemple

σ_{Δ}= (4,5,3,<u>1</u>,3,4,5,6,<u>2</u>,5,6)



- Problème de minimisation du nombre de travaux en retard:

$\Rightarrow$déterminer une séquence admissible pour <u>une sélection E*$\subset$V la plus grande possible</u>

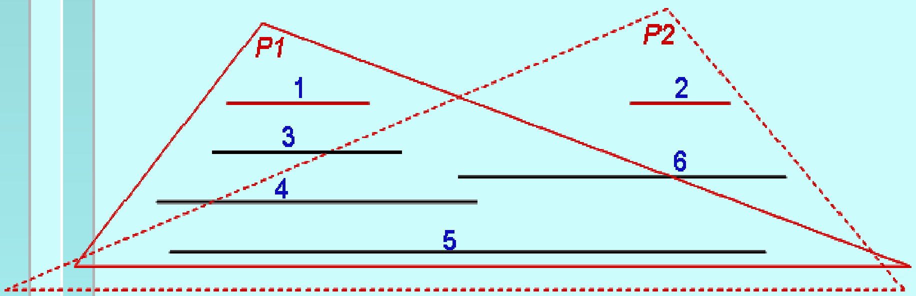
Corollaire 1. L'union de toutes les séquences dominantes que le théorème des pyramides caractérise pour toute sélection de travaux est dominante vis-à-vis du critère.

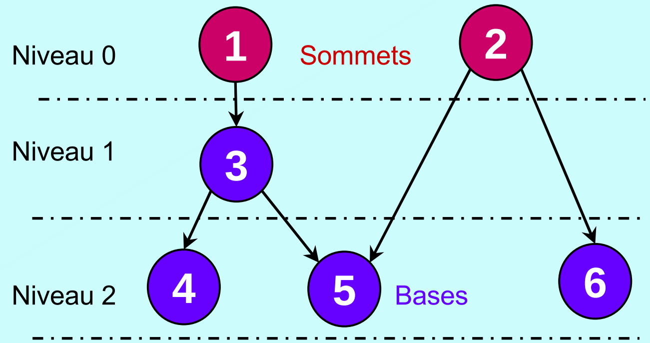
Problème à une machine

Dominance du théorème des pyramide

Problème à une machine avec fenêtres d'exécutions.

Un graphe d'intervalles sommet-base





Le graphe d'intervalles sommet-base

- ❑ Notion de sommet générateur
- ❑ Sous l'hypothèse qu'il n'existe aucun sommet ou ensemble de sommets générateurs avec bases communes: Le théorème des pyramides est dominant vis-à-vis du problème de minimisation du nombre de travaux en retard.
- ❑ Cas général :Un ensemble de séquences dominantes pour le problème peut ainsi être caractérisé par énumération de plusieurs séquences maîtres-pyramides.

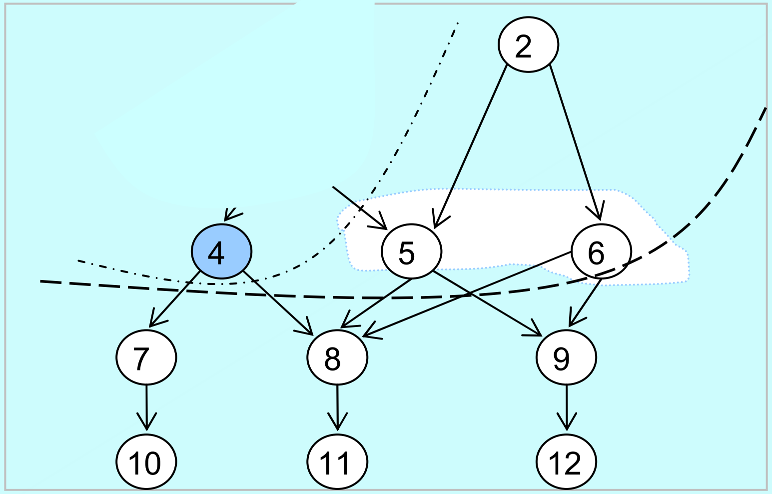
Problème à une machine

Dominance du théorème des pyramides: Exemple

- **Exemple**

  $V=\{1,2,\ldots,12\}$,

  3 : sommet générateur



Solution optimale

σ_Δ: *SMP de V={0,1,…,12}*

12, 9, 5, 3, <u>1</u>, <u>3, 4, 7, 10, 8, 11, 5, 9, 12, 6,</u> <u>2</u>, 5, 8, 11, 6, 9, 12

0 ou 1 job en retard

10, 7, 11, 8, 4

1,3 absents

7, 10, <u>8, 11, 12, 9</u>, 5, 6, 2, 5, 8, 11, 6, 9, 12

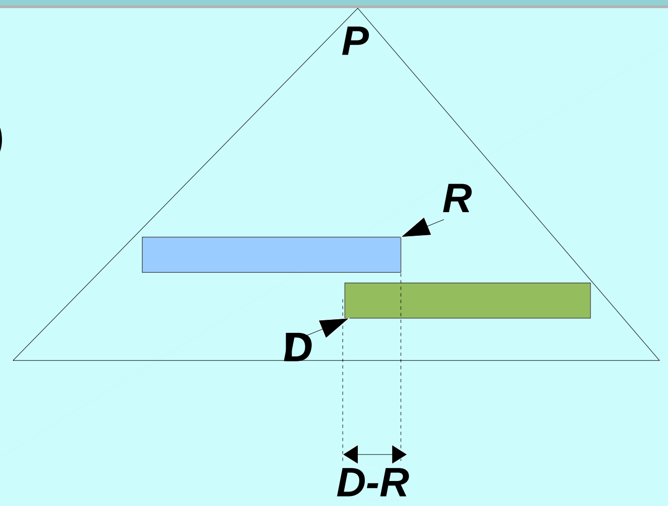
Plus de 2 jobs en retard

σ'_Δ: *SMP de V\\{1,3}*

Problème à une machine: Modélisation

Critères: *Admissibilité et Nombre de travaux en retard*

- **Cas d'une pyramide parfaite**



$$\sigma_{\Delta}=(n,n-1,\ldots,1,0,1,\ldots,n-1,n)$$

$x_i^- \leftarrow 1$: *i* séquencé à droite du sommet
$x_i^+ \leftarrow 1$: *i* séquencé à gauche du sommet

$$R(i)=r_i+\sum_{j=0}^{i} p_j - \sum_{j=1}^{i} p_j x_j^-$$
$$D(i)=d_i-\sum_{j=0}^{i} p_j + \sum_{j=1}^{i} p_j x_j^+$$

Admissibilité

$$\begin{aligned}
&\text{Max} \quad z = D - R \\
&R \geq R(i) && (1.1)\\
&D \leq D(i) && (1.2)\\
&x_i^+ + x_i^- = 1 && (1.3)\\
&x_i^+, x_i^- \in \{0,1\} \\
&D, R \in Z \quad \forall i = 0 \cdots n
\end{aligned}$$

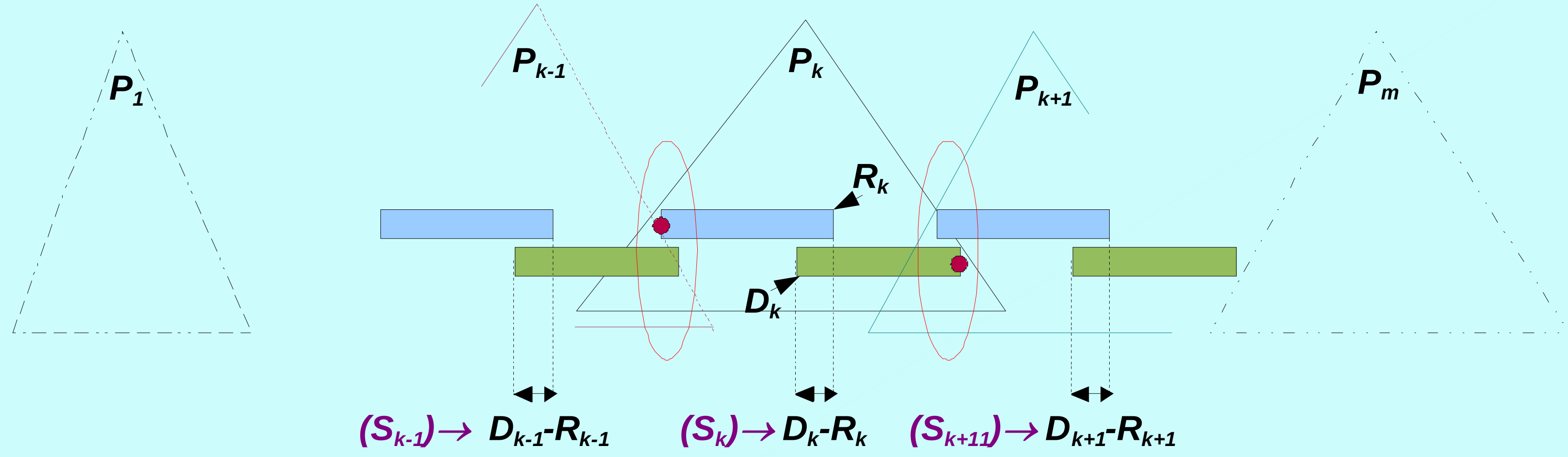
Nombre de jobs en retard

$$\begin{aligned}
&\text{Min} \quad w = \sum_{i=1}^{n} x_i^+ + x_i^- \\
&R \geq R(i) \\
&D \leq D(i) \\
&x_i^+ + x_i^- \geq 1 && (1.4)\\
&D - R \geq -p_t && (1.5)\\
&x_i^+, x_i^- \in \{0,1\}; D, R \in Z; \quad \forall i
\end{aligned}$$

Problème à une machine: Modélisation

Critères: *Admissibilité et Nombre de travaux en retard*

- **Cas Multi-pyramides**



$$eft_k = R_k + \sum_{i=1}^{n} x_{ik}^{-} p_{ik} \quad (i.1)$$

$$lst_k = D_k - \sum_{i=1}^{n} x_{ik}^{+} p_{ik} \quad (i.2)$$

$$s_{ik} = \max(r_{ik}, eft_{k-1}) \quad (i.3)$$

$$f_{ik} = \min(d_{ik}, lst_{k+1}) \quad (i.4)$$

$$\text{Min } z = \Delta$$

$$R_k \geq \max(r_{ik}, eft_{k-1}) + \sum_{j=0}^{i} p_{jk} - \sum_{j=1}^{i} p_{jk} x_{jk}^{-} \qquad (2.1)$$

$$D_k \leq (d_{ik}, lst_{k+1}) - \sum_{j=0}^{i} p_{jk} + \sum_{j=1}^{i} p_{jk} x_{jk}^{+} \qquad (2.2)$$

$$\Delta \leq D_k - R_k + p_{tk} \qquad (2.3)$$

$$\sum_{k=u(j)}^{v(j)} x_{jk}^{+} + x_{jk}^{-} = 2\big(v(j) - u(j)\big) + 1 \qquad (2.4)$$

$$x_{ik}^{+}, x_{ik}^{-} \in \{0,1\}; \quad D_k, R_k \in Z; \quad \forall i = 1 \cdots n_k, \forall k = 1 \ldots m$$