Contrainte de flot pour RCPSP avec temps de transfert

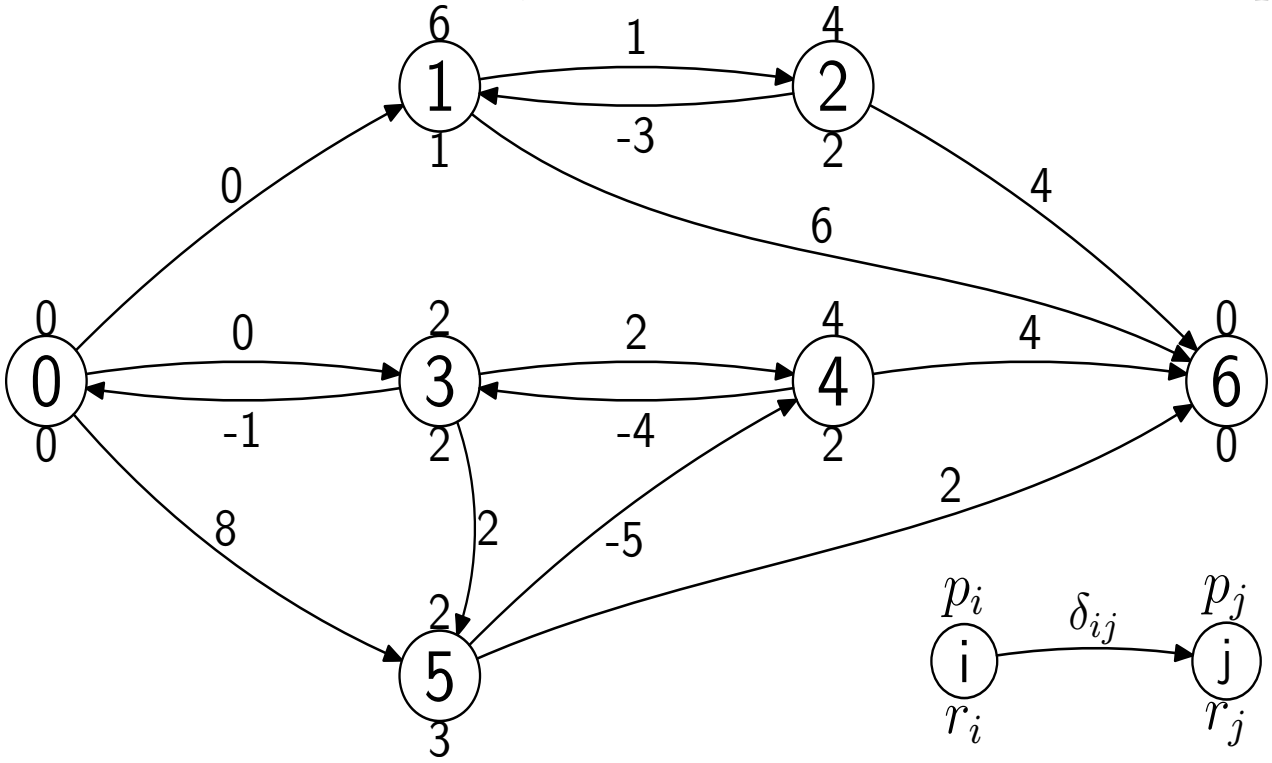
$$PS|temp, s_{ij}|Cmax$$

- Benoist Thierry BOUYGUES/e-Lab
- **Diamantini Maurice** ENSTA/LMA

Présentation du x-RCPSP

exemple mono-ressource

Pour $n = 5$ activités, 1 seule ressource en 3 exemplaires



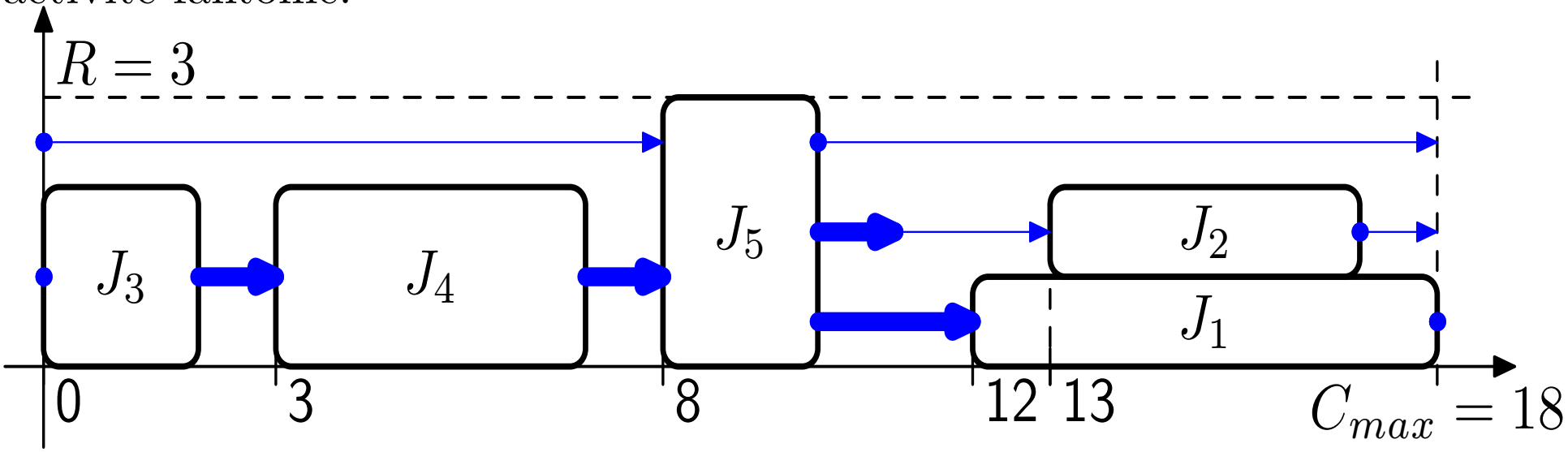
Graphe de précédence

	1	2	3	4	5
1	.	3	3	1	7
2	0	.	1	1	4
3	2	1	.	1	6
4	1	2	0	.	1
5	2	1	2	1	.

Temps de "setup"

Solution de l'exemple

Les temps de transfert «*consomme*» de la ressource au même titre qu'une activité fantôme.



Peut-on placer J_1 devant J_5 ?

- mais une contrainte de précédence négative l'interdit ($\delta_{12} = [1, 3]$)
- et le temps de transfert ($v_{15} = 4$) repousserait J_5 de J_1

Sommaire

Contrainte globale de flot (CGF)

Principe :

- entrée :
  - un graphe orienté avec des capacités min/max sur les arcs
  - nœuds sources et puits et une variable pour le flot global (R^k)
- la consistance :
  - propage la lois de Kirchoff au niveau de chaque nœuds
  - garantit l'existence d'un flot en maintenant un flot support
    (pas nécessairement réalisable du point de vue des autres contraintes)

Application au x-RCPSP :

- une CGF pour chaque ressource ($\Rightarrow m$ contraintes de flot)
- chaque activité i est représentée par 2 nœuds et un
  arc interne étiquetée par la demande r_i^k
- les graphes de ressources sont complets
  (contrairement à l'unique graphe de précédence trés peu dense)

Changement de variables : les variables d'écart

On observe que $S_j - S_i$ intervient directement dans deux contraintes qui régissent l'ordonnancement :

–	précédence	$\forall i, \forall j$	$S_j - S_i \geq \delta_{ij}$
–	temps de transfert	$\forall i, \forall j, \ \forall k$	$\phi_{ij}^k > 0 \Rightarrow S_j - S_i \geq p_i + v_{ij}^k$

On crée donc des variables auxiliaires : les variables d'écart $e_{ij} = S_j - S_i$
$\Rightarrow$ S_i devient le cas particulier : $S_i = e_{0i}$
$\Rightarrow$ nouvelles contraintes pour la sémantique : $e_{ij} = e_{0j} - e_{0i}$ (transitivité)

Conséquences :

- AV : meilleure propagation de la contrainte de transfert (couplage direct entre les variables ϕ_{ij}^k et e_{ij})
- AV : fournit un levier lors de l'énumération pour agir sur la position relative de deux activités
- INC : n^2 variables au lieu de n (pas d'importance)

Modèle adopté pour la PPC

Objectif : minimiser la date de fin du projet : $C_{\text{MAX}} = \min_{\phi_{ij}^k, e_{ij}} e_{0,n+1}$

Décisions : e_{ij} écart des dates de démarrage entre les activités i et j,
ϕ_{ij}^k flux inter-activités (i,j) pour la ressource k,

Contraintes :

– précédence	$\forall i, \forall j$	$e_{ij} \geq \delta_{ij}$	(n^2)
– temps de transfert	$\forall i, \forall j,\ \forall k$	$\phi_{ij}^k > 0 \Rightarrow e_{ij} \geq p_i + v_{ij}^k$	$(n^2.m)$
– flot (CGF) :	$\forall k,$	un flot de valeur R^k est possible	(n)
– transitivité :	$\forall i, \forall j$	$\mathrm{e_{ij}} = \mathrm{e_{0j}} - \mathrm{e_{0i}}$ $(= S_j - S_j)$	(n^2)
– cumulative :	à chaque date t, la somme des ressources k utilisées par toute activité i est limitée à R^k		(1)

Enumération : principe général

Constatation : deux variables de décision : ordo e_{ij} et flots ϕ_{ij}^k

- à ordonnancement fixé (les S_i ou e_{ij}) la détermination d'un flot est facile car seule les variables de flot autorisées sont non nulles par $C_{\text{transfert}}$
- à flot fixé (les ϕ_{ij}^k) le calcul de l'ordonnancement est facile ($\approx$ PERT) car de nombreuses précédences sont imposées par $C_{\text{transfert}}$

Sur quelles variables brancher ?

- seul l'ordo intervient dans l'objectif (S_{n+1}),
- beaucoup de flots différents conduisent à un même ordo

Par conséquent :

- on basera l'énumération sur la recherche d'un ordonnancement,
- on n'utilisera les flots que comme une aide.

et maintenant, comment construire ce ordonnancement ?

Enum. phase 1 : profil réalisable

Principe : rendre le profil au plus tôt «*satisfaisant*» pour chaque ressource (le profile ignore les temps de transfert)

1. on recherche la première date de sur-utilisation de ressources (plusieurs conflits simultanés sont possibles)
2. on extrait toutes les activités participant à ces conflits
3. on va avancer une activité j en conflit par rapport à une autre activité i du même conflit en agissant sur la variable d'écart e_{ij}

Nota :
les deux activités à écarter doivent faire partie d'un même conflit à cause des précédences généralisées (sinon risque de déplacement en bloc du conflit).

Enum. phase 1 : choix des variables

Décision de branchement : écarter deux activités j et i d'un même conflit.

Oui mais sur quels critères ?

- traiter tous les conflits simultanés ensemble ou séparément ?
- quelle activité j repousser ?
- par rapport à quelle activité source i ?
- prendre en compte, pour une activité :
  - le nombre de ressources demandées ?
  - le nombre de conflit simultanés ?

Notre choix :

- extraction de tous les arcs (i, j) candidats à l'écart, donc les e_{ij}
- sélection de l'arc par évaluation de plusieurs critères
  - LST = LatestStartTime pour j $\surd$
  - ECT = EarliestCompletionTime pour i $\surd$
  - exploiter les supports de flot sur l'arc ? $\surd$

Enum. phase 1 : exploitation du flot support

Comment tenir compte des informations sur les flots dans le choix des e_{ij} ?

- La contrainte $C_{\text{transfert}}$ est le seul lien entre ordo et flot
  - si $\phi^{\kappa}_{ij} > 0$ alors e_{ij} est orienté «+»
  - si e_{ij} est orienté «+» alors $\phi^{*}_{ji} = 0$
- le support φ^{k}_{ij} est une proposition pour chaque ϕ^{k}_{ij} de la CGF

donc **le support fournit localement** pour en chaque arc
une information globale sur le flot courant

Comment en tenir compte dans le choix de l'arc à «*écarter*» ?

- favoriser les arcs e_{ij} ayant le plus grand cumul des supports $\sum_k \varphi^{k}_{ij}$
- favoriser les arcs e_{ij} ayant le plus grand nombre de supports positifs $\surd$
- pondérer ces critères par une «*crédibilité*» de chaque support φ^{k}_{ij}

Enum. phase 2 : figeage au plus tot

Principe :

- on branche sur les dates de démarrage $S_i = e_{0j}$
- en choisissant l'activité la plus précoce,
- qu'on essaie de placer au plus tôt

d'où le branchement

- branche 1 : $S_i := \inf(S_i)$ puis propagation,
- branche 2 : $\inf(S_i) := \inf(S_i) + 1$ puis propagation.

$\Rightarrow$ si le profil est remis en cause on revient en phase 1

Filtrage par les flots

Problème :

– $C_{\text{transfert}}$ longtemps inefficace en cours d'énum car $\inf(\phi_{ij}^k)$ souvent nulle

Définitions :

– soit les producteurs réels de ressource k pour j : $P_k(j) = \{i \mid \phi_{ij}^k > 0\}$
– soit les producteurs potentiels : $\mathcal{P}_k(j) = \{i \mid \sup(\phi_{ij}^k) > 0\}$

Solution : filtrage par les producteurs potentiels

Pour chaque tâche i et chaque ressource k :

– on classe les $i \in \mathcal{P}_k(j)$ par date de libération croissante de la ressource k
– on «*remplit*» j par le $\sup(\phi_{ij}^k)$ de caque prod. potentiel i
– le premier i^* qui «*sature*» j donne une borne inférieure du début de j

$$\inf(S_j) \geq \inf(S_{i^*}) + p_{i^*} + v_{i^*j}^k$$

Tests et résultats

Conditions de test

- environnement java de PPC choco.sourceforge.net
- avec contrainte globale de flot [E. Gaudin, G. Rochard]
- avec contrainte globale cumulative [N. Beldiceanu]

Instances utilisées

- instances de base ubo10, ubo20, ubo50 (5 ress.) web : "ProGenMax"
- avec ajout des temps de transfert par [Christoph Schwindt]
  livre : "Project Scheduling with time windows and scarse resources"

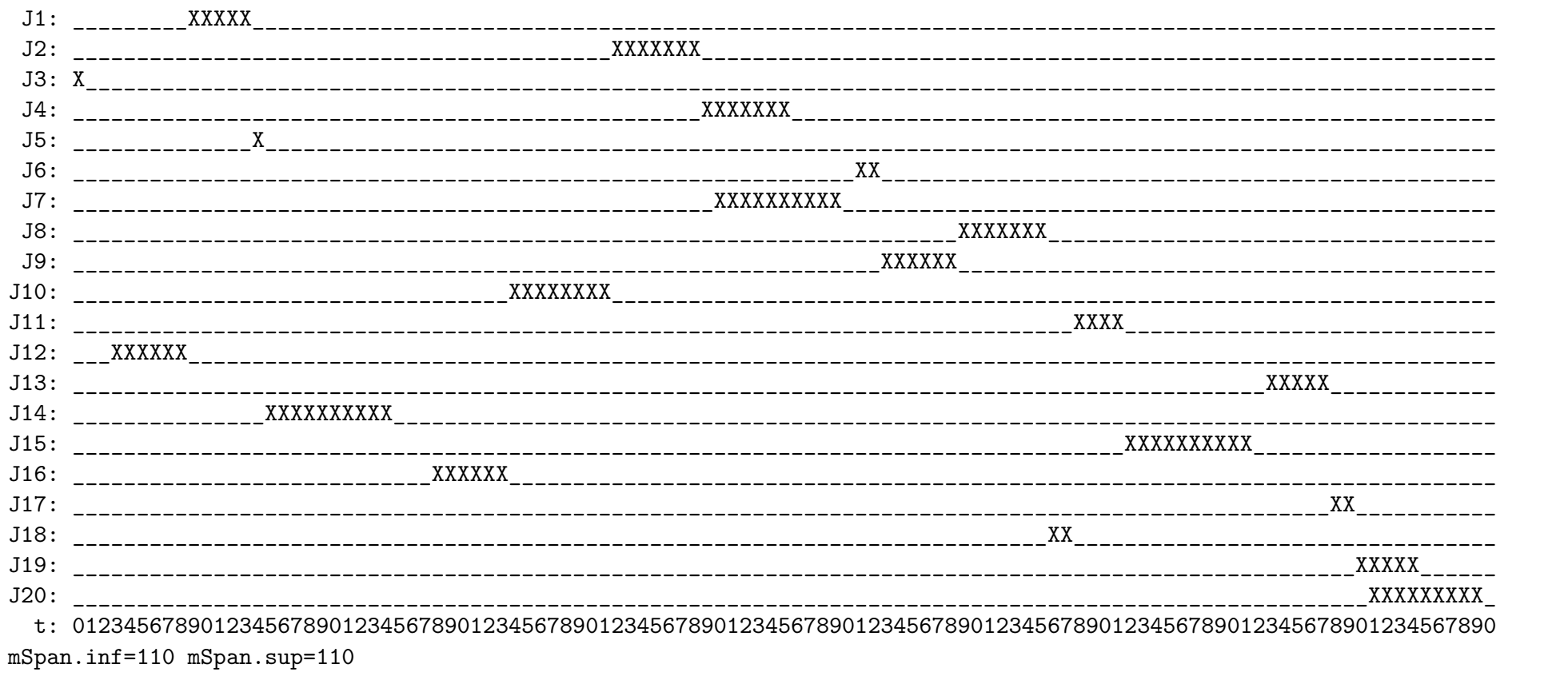
Résultats

- on s'est concentré sur des petites instances non résolues (10*, 20 et 50)
- valeurs connues (5 ans) souvent confirmées, améliorées voire prouvées
- ... mais les résultats peuvent prendre des heures,
- ... et les critères de branchement restent à affiner

* dans le livre em Project Scheduling... 2nd édition, toutes les instances ubo10 sont considérées résolues

Exemple de résultat

instance : ubo_20_10_1 (5 ress.) valeur connue : 111, nous : 110*
critère de choix des arcs : LST = ECT = supportFlow (⟶ sensible)
durée trouvée en 2mn, prouvée en 1h40 (station linux bipro)



Conclusion et perspectives

Conclusion

- La contrainte de flot décharge pas mal de travail en rendant quasi automatique la prise en compte des temps de transfert,
- l'utilisation du support : trés prometteur, mais à affiner,
- cette approche est compétitive, avec levier supplémentaire

Perspectives

- affiner les critère de choix des arcs, en particulier les conditions d'exploitation du support de flot,
- intégrer les critères classiques de branchement dans notre énumération,
- recherche d'heuristiques ou exploration incomplète exploitant les flots pour les instances plus importantes.