

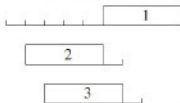
LA PROGRAMMATION LINÉAIRE POUR DES PROBLÈMES D'ORDONNANCEMENTS

C. DÜRR (Lix)

avec Mathilde Hurand

Exemple $P|r_j; p_j=p; d_j|\sum C_j$

- P on a m machines identiques à notre disposition
 $p_j=p$ chacune des n tâches a la même durée p
 r_j, d_j chaque tâche j vient avec un intervalle $[r_j, d_j)$
où l'exécution est autorisée
 $\sum C_j$ minimiser le temps de complétude moyen



$$m=2, \quad p=4, \quad r = 1, 2, 3 \\ d = 10, 7, 8$$

HISTORIQUE

$O(m n^2)$
par un algo
compliqué

Barbara Simons
Manfred Warmuth

1989
SIAM J. Comp.

par progr.
linéaire

Peter Brucker 2005
Svetlana Kravchenko
Journal of Scheduling

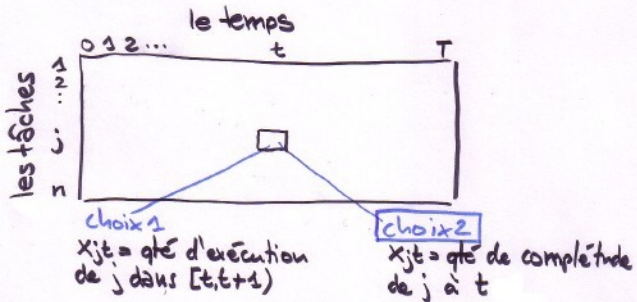
$O(n^4)$
simple
par rech.
plus courts
chemins

Mathilde Hurand 2006
C.D.

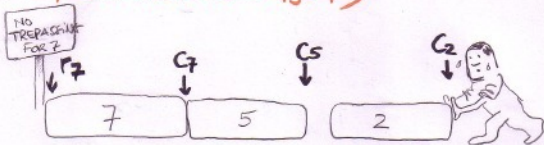
[arXiv.org > cs > cs/0602016](https://arxiv.org/abs/cs/0602016)

MODÉLISATION PAR UN PROGR. L'IN.

LES VARIABLES



DIMINUER LA GRANULARITÉ DU TEMPS (DÉTAIL, STANDARD POUR $p_j = p$)

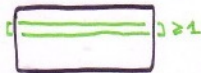


on veut minimiser $\sum C_j$
dans une solution optimale

$$\forall j \quad C_j \in \{r_i + a \cdot p : i \in [1, n], a \in [1, n]\}$$

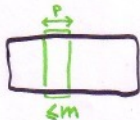
on peut se restreindre à n^2 temps
(time points)

AJOUTER DES CONTRAINTES



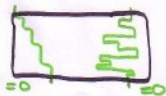
chaque tâche doit compléter

$$\forall j \quad \sum_t x_{jt} \geq 1$$



au plus m tâches peuvent exécuter en même temps

$$\forall t \quad \sum_j \sum_{t_s \leq t < t+p} x_{js} \leq m$$



les exécutions doivent être dans les intervalles autorisés

$$\forall j \quad \forall t \notin [r_j + p, d_j] \quad x_{jt} = 0$$

LA QUESTION

QUE JE ME POSE

Qui possède depuis 1 an

SOIT LE PROGRAMME
LINÉAIRE

$$\min C^T x$$

t.q.

$$Ax \leq b$$

$$Bx \geq d$$

A, B matrices 0,1

b, d entier

pour certains C
il y a toujours une
solution optimale
entière.

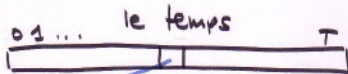
comment le montrer?

Si x est une solution

entière, on peut trouver

une solution optimale

UN AUTRE PROGRAMME LINÉAIRE



y_t = nb d'emplacements (slots)
qui terminent à t ou avant

LES CONTRAINTES

$$y_0 = 0$$

$$(y_T = n)$$

$\forall t$

$$y_{t-1} \leq y_t$$

$$y_{t+p} - y_t \leq m \quad [\text{chargement}]$$

et [inclusion] ... transp. suivant

FONCTION OBJECTIVE

$$\min \sum t (y_t - y_{t-1})$$

$$= \min \sum -y_t \equiv \max \sum y_t$$

EXPLICATION INCLUSION

$$\forall i, j \quad s = r_i + p \quad y_t - y_s \geq c_{ij} \quad [\text{inclusion}]$$

$t = D_j$

$$c_{ij} = |\{k : [r_k, D_k] \subseteq [r_i, D_j]\}|$$



il faut suffisamment
d'emplacements
pour toutes les
tâches qui
doivent s'exéc.
dans $[r_i, D_j]$

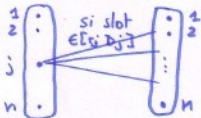
QUE FAIRE MAINTENANT ?

LEMME Toute solution
entière corresp. à
un ordonnancement
équivalent

LE PROGR. LINÉAIRE a
une forme connue :

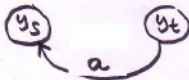
$$y_t - y_s \leq a$$

tâches emplacements



on applique (inclusion)
au théorème de Hall
pour montrer l'exist.
d'un couplage.

on définit un graphe $G(V, E)$
 V = variables
 E = contraintes



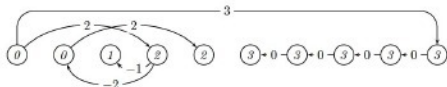
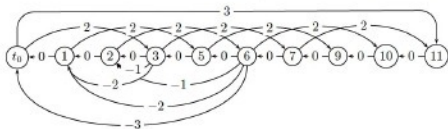
L'INTERPRÉTATION DU GRAPHE

chaque chemin de poids a de y_0 à y_t , correspond à l'inégalité

$$y_t - y_0 \leq a \quad \equiv \quad y_t \leq a$$

le plus court chemin correspond à l'inégalité la plus forte.

Poser $y_t = \text{distance}(y_0, y_t)$ est une solution optimale



CONCLUSION

- Est-ce qu'on peut trouver un meilleur algo. de plus courts chemins pour ce graphe particu?

autre chose

- Trouver des bons problèmes pour contribuer au projet ALPAGE.

