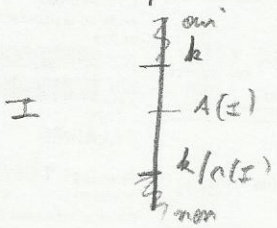
Remarque : On étend la définition de NP-hard pour les pb de gap.

Théorème : Si gap$_\rho$ NP-hard, alors pas de ρ'-approx avec $\rho' < \rho$ (sauf si P=NP).

PROOF : Supposons que l'on ait A (polynômial), une ρ'-approx, $\rho' < \rho$ $\Rightarrow$ comment résoudre gap$_\rho$?



On calcule $A(I)$. Si $A(I) < k \Rightarrow OPT(I) \leq A(I) < k \Rightarrow$ NO

Sinon $\rho'(I)\, OPT(I) \geqslant A(I) \geqslant k \Rightarrow OPT(I) \geqslant \frac{k}{\rho'(I)} \overset{\text{si } \rho' < \rho}{>} \frac{k}{\rho(I)} \Rightarrow$ on dit "yes" $\square$.

Exos : gap introduct°, reduct° bin packing
gap introduct°, reduct°

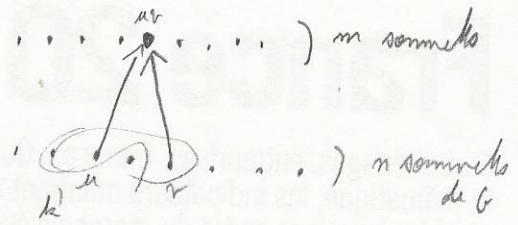
Def :
- gap introducing reduction : réduct° d'un pb de décision vers un pb gap.
- gap preserving reduction : réduct° depuis un pb gap vers un pb gap.

Théorème : gap $4/3$ est NP-hard pour $P|prec|C_{max}$.

PROOF : réduction depuis clique

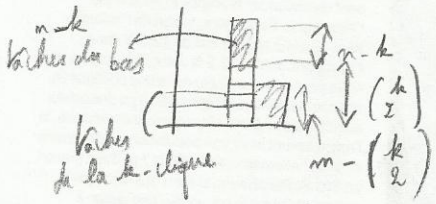
π_{clic} : given $G=(V,E)$, clique $\geqslant k \Rightarrow$ oui
clique $\leq k-1 \Rightarrow$ non

Soit $G' = (DAG)$



- $n' = n+m$
- $m' = 2m$
- m_a (# machines)
- pour chaque $uv \in E$

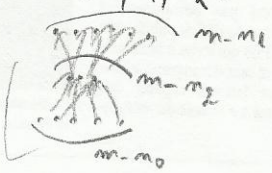
Si k cliques, les k tâches ordonnancées à $t=0$ libèrent $\binom{k}{2}$ tâches pour le temps 1.



Définition de l'instance : Soit $n_0 = k$, $n_1 = \binom{k}{2} + (n-k)$
$n_2 = m - \binom{k}{2}$

Soit $m_a = \max(n_0, n_1, n_2) + 1$

Soit $p_{ij} =$



$\forall j,\ p_j = 1$

Montrons que clique $\geqslant k \Rightarrow OPT \leq 3$.

↑ donne

clique $\leq k-1 \Rightarrow OPT > 4$ (car opt $\leq 3 \Rightarrow$ clique $\geqslant k$) DANS TOUTE sol ≤ 3, on est obligés de faire ?
(no idle $\Rightarrow$) On est obligés de choisir k tâches du bas :