Algorithmique Avancée
Approximation – Problèmes de partitions

Denis TRYSTRAM
Notes de cours ENSIMAG Alternants 2A

Oct. 2023

La base : 2Partition

Version classique :

- 2Partition
- **Instance** : n entiers n_i et un entier pair $S = \Sigma_{1 \leq i \leq n} n_i$
- **Question** : Existe-t-il une partition des entiers en deux sous-ensembles A_1 et A_2 telle que $\Sigma_{i \in A_1} n_i = \Sigma_{i \in A_2} n_i$?

La base : 2Partition

Version classique :

- 2Partition
- **Instance** : n entiers n_i et un entier pair $S = \Sigma_{1 \leq i \leq n} n_i$
- **Question** : Existe-t-il une partition des entiers en deux sous-ensembles A_1 et A_2 telle que $\Sigma_{i \in A_1} n_i = \Sigma_{i \in A_2} n_i$?

Sous forme d'optimisation

On cherche ici à réduire l'écart entre deux partitions d'entiers. C'est donc un problème de minimum.

- Notons ω la somme des entiers de la plus grande des deux partitions.
- n_{max} est l'entier le plus grand de l'instance.

Approximation de 2Partition

- On considère l'algorithme glouton qui remplit à tour de rôle chaque sous-ensemble en mettant l'entier courant dans le sous-ensemble le moins chargé.

Propriété

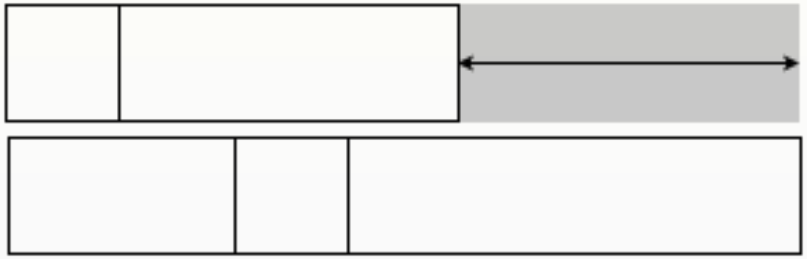
Cet algorithme a une approximation garantie.

Analyse

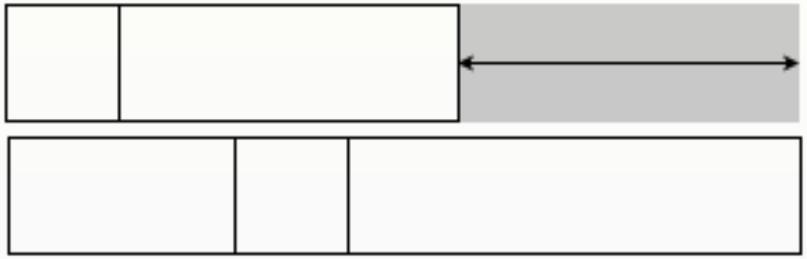
L'analyse s'obtient à partir de deux **bornes inférieures**[1] de la solution optimale :

- On ne peut faire mieux qu'une découpe parfaite
  $\omega^* \geq \frac{\Sigma_j n_j}{2}$
- On ne peut faire moins que le plus grand entier
  $\omega^* \geq n_{max}$

[1] c'est un problème de minimisation

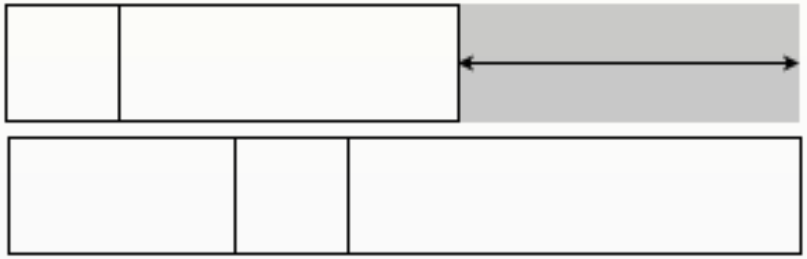


$$2 \cdot \omega = \Sigma_j n_j + S_{idle}$$



$$2 \cdot \omega = \Sigma_j n_j + S_{idle}$$

- $\omega = \frac{\Sigma_j n_j}{2} + \frac{1}{2} n_{max}$
- $\omega \leq \omega^* + \frac{1}{2}\omega^*$



$$2 \cdot \omega = \Sigma_j n_j + S_{idle}$$

- $\omega = \frac{\Sigma_j n_j}{2} + \frac{1}{2} n_{max}$
- $\omega \leq \omega^* + \frac{1}{2}\omega^*$

On obtient ainsi une $\frac{3}{2}$-approximation de l'optimum.

Tightness

La question ici est d'étudier si la borne d'approximation précédente est la meilleure que l'on puisse obtenir.

On considère l'instance suivante :

- 3 entiers : 1,1 et 2.
- Optimal : $\omega^* = 2$ (les deux "1" d'un coté et le "2" de l'autre)
- Glouton partionne avec "1" d'un coté et "1" suivi de "2" de l'autre,
  donc, $\omega = 3$

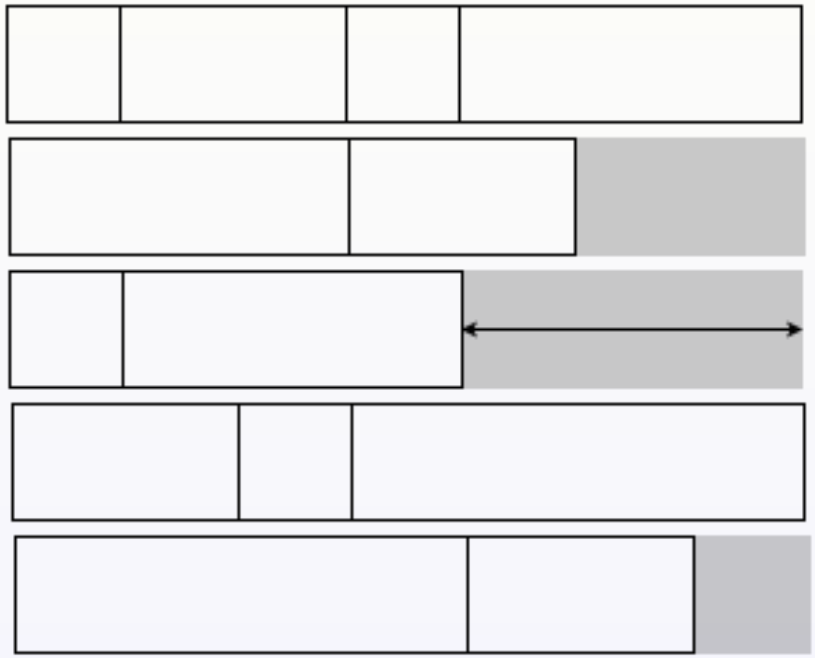
Le pire cas est donc atteint par cette instance particulière.

Extension : partitionnement sur m ressources

On considère le même problème mais avec $m \geq 3$ partitions.

- Ordonnancement Tâches Indépendantes
- **Instance** : n entiers n_i, un entier m et un entier K
- **Question** : Existe-t-il une m-partition des entiers en moins que K ?

L'analyse est obtenue en généralisant l'argument de surface sur le diagramme ressources/temps.



$$m \cdot \omega_{LS} = W + S_{idle} \text{ (où } W = \Sigma_j n_j)$$

Analyse

Proposition.

List scheduling (LS) est une 2-approximation.

Comme précédemment, la preuve est basée sur deux bornes inférieures :

- $\omega^* \geq \frac{W}{m}$
- $\omega^* \geq n_{max}$

Analyse

Proposition.

List scheduling (LS) est une 2-approximation.

Comme précédemment, la preuve est basée sur deux bornes inférieures :

- $\omega^* \geq \frac{W}{m}$
- $\omega^* \geq n_{max}$

$$m \cdot \omega_{LS} = W + S_{idle}$$

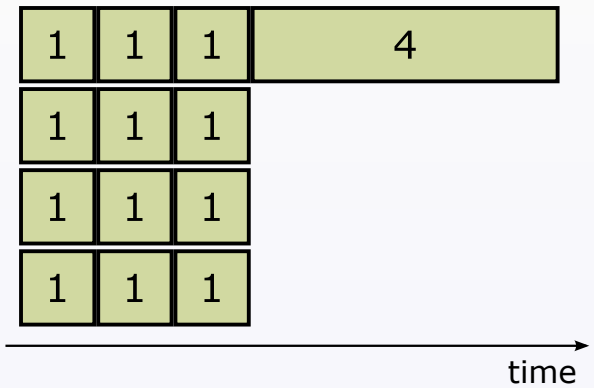
$$S_{idle} \leq (m-1) \cdot n_{max} \leq (m-1) \cdot \omega^*$$

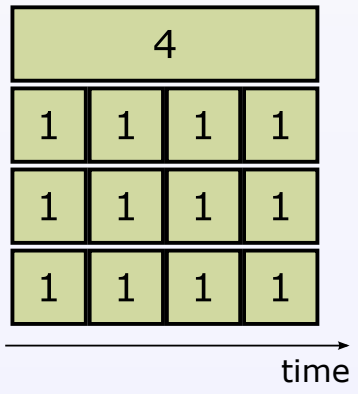
$$\omega_{LS} = \frac{W}{m} + \frac{S_{idle}}{m} \leq (1 + \frac{m-1}{m}) \cdot \omega^*$$

Est-ce le meilleur résultat possible ?

Propriété

La borne de pire cas de $2 - \frac{1}{m}$ est atteinte[2].

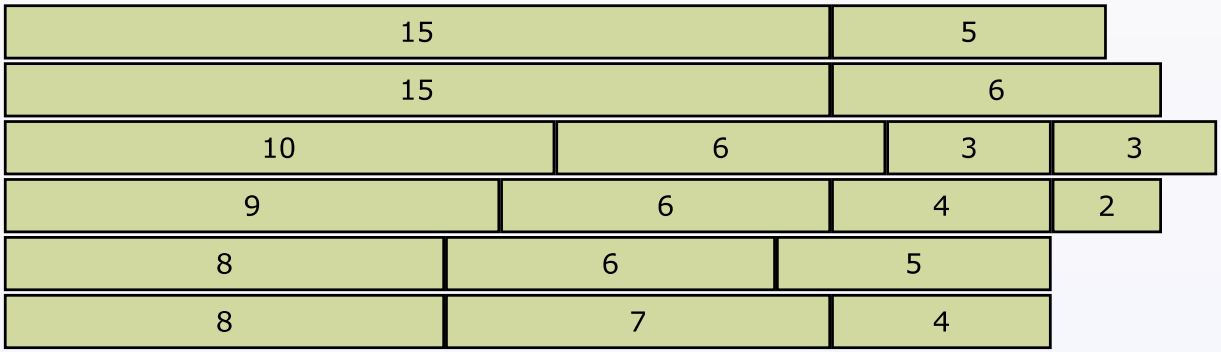




[2]on montre pour $m = 4$ mais le résultat est général

Amélioration : la règle LPT

Partant de l'analyse du pire cas, il est possible d'améliorer la borne d'approximation en considérant la politique LPT (Largest Processing Time).



Approximation: $\frac{3}{2}$ (pour $m \geq 2$)