Algorithmique Avancée
Approximation – Partition(s)

Denis TRYSTRAM
Notes de cours ENSIMAG Alternants 2A

Oct. 2021

Un autre exemple : 2Partition

Version classique (sous forme de décision) :

- 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 $\Sigma_{i \in A_1} n_i = \Sigma_{i \in A_2} n_i$?

Un autre exemple : 2Partition

Version classique (sous forme de décision) :

- 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.
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

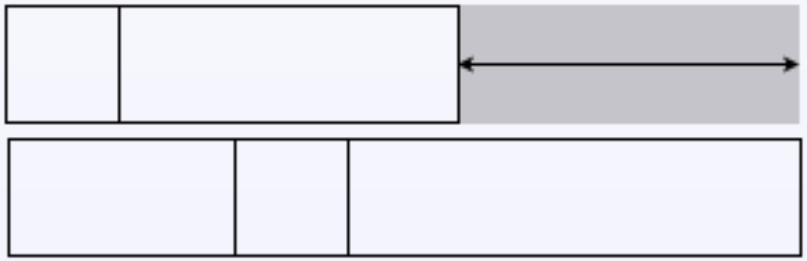
L'analyse s'obtient à partir de deux bornes inférieures de la solution optimale :

- 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
  $\omega^* \geq n_{max}$

Analyse

L'analyse s'obtient à partir de deux bornes inférieures 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
  $\omega^* \geq n_{max}$



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

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

Tightness

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

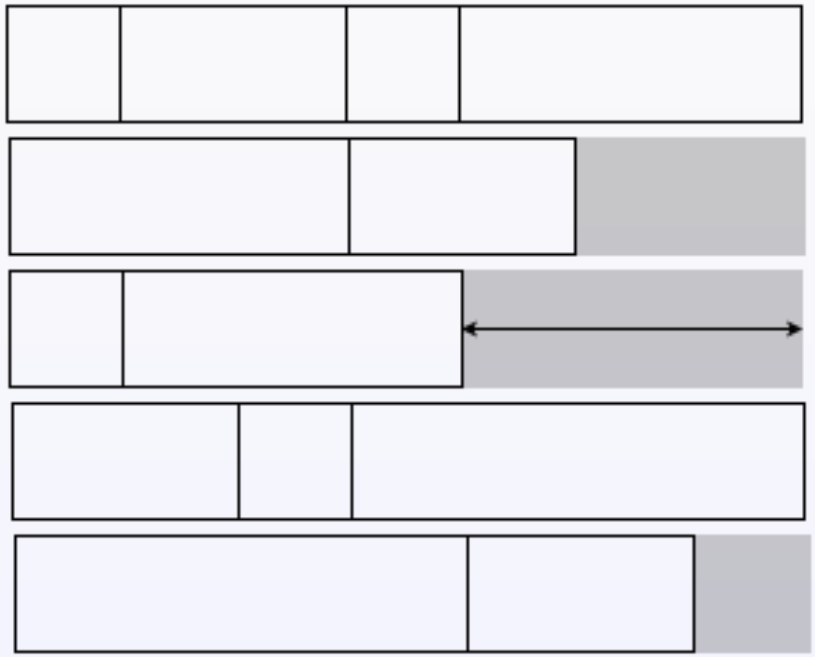
On considère l'instance suivante :

- 3 entiers 1,1 et 2.
- Optimal : $\omega^* = 2$
- Glouton partionne avec 1 d'un coté et 1 et 2 de l'autre, donc, $\omega = 3$

Extension : partitionnement sur m

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

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



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

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

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

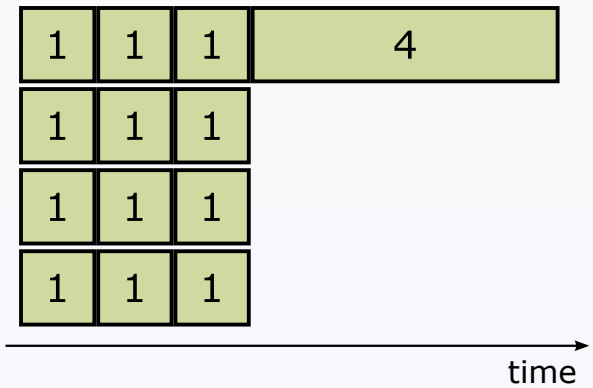
$$S_{idle} \leq (m-1) \cdot n_{max} \leq (m-1) \cdot \omega^*$$

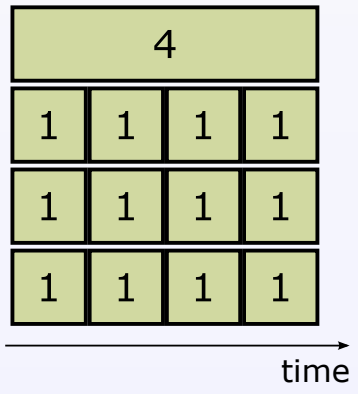
$$\omega_{LS} = \frac{W}{m} + \frac{S_{idle}}{m} \leq (1 + \frac{m-1}{m}) \cdot \omega^*$$

Tightness

Propriété

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

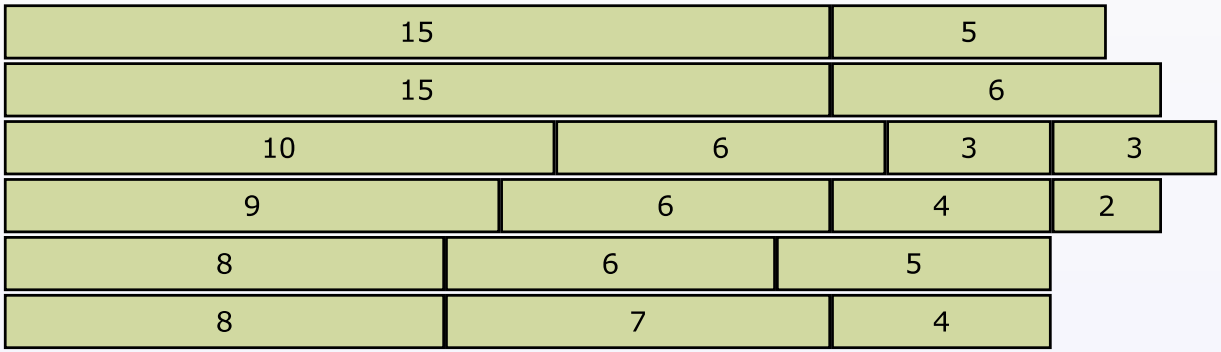




[1] on montre pour $m = 4$ mais le résultat est général

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 $m \geq 2$)

Un autre exemple dual : le BinPacking

- BinPacking
- **Instance** : Un ensemble de *bins* identiques de taille H
  un ensemble d'entiers s_i pour $1 \leq i \leq n$
- **Question** : Ranger tous les entiers dans un minimum de bins.

On note N_A le nombre de boites requis par l'algorithme A.

Approximation

On considère l'algorithme First Fit (FF) qui place les objets les uns au dessus des autres dès qu'ils rentrent.

Approximation

On considère l'algorithme First Fit (FF) qui place les objets les uns au dessus des autres dès qu'ils rentrent.

- Supposons que l'algorithme FF utilise m boites, alors, au moins $m-1$ boites sont remplies à plus de la moitié (sinon, l'algorithme aurait placé ces objets dans la même boite).
- La preuve est alors immédiate en utilisant cet argument : $N^* \geq \Sigma_{1 \leq i \leq n} s_i > \frac{m-1}{2}$

  Comme FF utilise m boites, on a bien $N_{FF} < 2.N^* + 1 \leq 2.N^*$