

*Sans musique la vie serait une erreur.*

— Friedrich Nietzsche



# Rangement d'objets multiboîtes

modèles et algorithmes

Pierre Lemaire



6 septembre 2004

*Ouverture*

The background of the slide features two staves of musical notation. The notation is rendered in a light gray color and is slightly faded. It consists of two staves, each with five lines. The notes are represented by black stems and heads, with some notes beamed together. The overall style is that of a classical score.

*Problèmes de rangements  
d'objets multiboîtes*

# Un problème à résoudre : l'analyse de documents

# Un problème à résoudre : l'analyse de documents



des documents

# Un problème à résoudre : l'analyse de documents



des documents



des experts

# Un problème à résoudre : l'analyse de documents



des documents



des experts

- chaque document doit être analysé par un expert

# L'analyse de documents : un problème de rangement



# L'analyse de documents : un problème de rangement

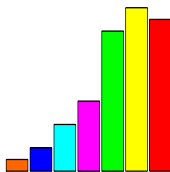


les documents

# L'analyse de documents : un problème de rangement



les documents

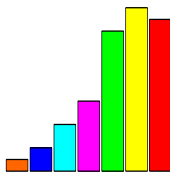


les objets

# L'analyse de documents : un problème de rangement



les documents



les objets

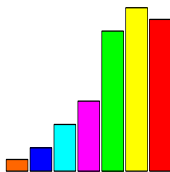


les experts

# L'analyse de documents : un problème de rangement



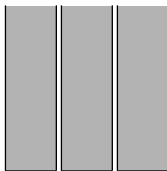
les documents



les objets



les experts

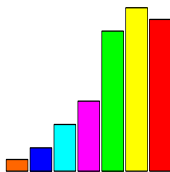


les boîtes

## L'analyse de documents : un problème de rangement



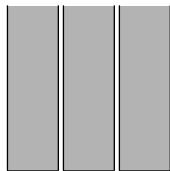
les documents



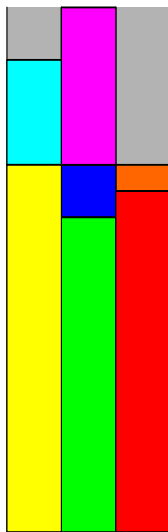
les objets



les experts



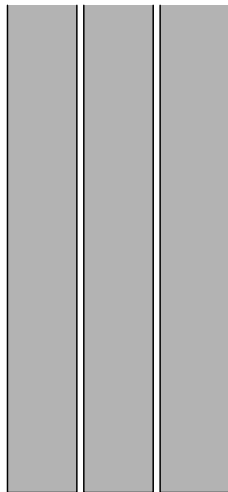
les boîtes



## Définition des problèmes de rangement

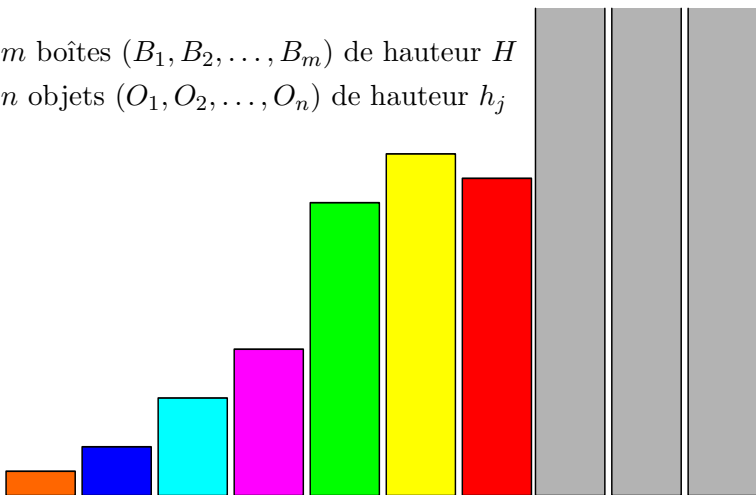
## Définition des problèmes de rangement

- $m$  boîtes  $(B_1, B_2, \dots, B_m)$  de hauteur  $H$



## Définition des problèmes de rangement

- $m$  boîtes ( $B_1, B_2, \dots, B_m$ ) de hauteur  $H$
- $n$  objets ( $O_1, O_2, \dots, O_n$ ) de hauteur  $h_j$



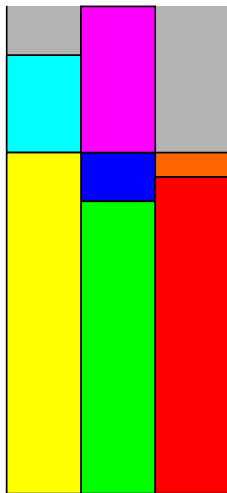


## Définition des problèmes de rangement

- $m$  boîtes ( $B_1, B_2, \dots, B_m$ ) de hauteur  $H$
- $n$  objets ( $O_1, O_2, \dots, O_n$ ) de hauteur  $h_j$

**question :**

peut-on ranger les  $n$  objets dans  
les  $m$  boîtes de hauteur  $H$  ?



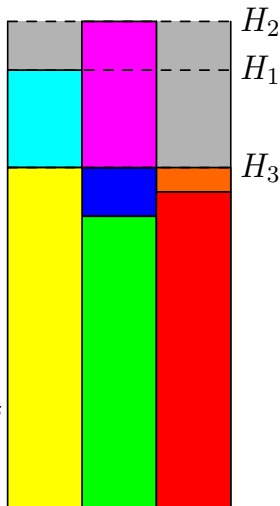
## Définition des problèmes de rangement

- $m$  boîtes ( $B_1, B_2, \dots, B_m$ ) de hauteur  $H$
- $n$  objets ( $O_1, O_2, \dots, O_n$ ) de hauteur  $h_j$

**question :**

peut-on ranger les  $n$  objets dans  
les  $m$  boîtes de hauteur  $H$  ?

- hauteur dans une boîte :  $H_i = \sum_{O_j \in B_i} h_j$



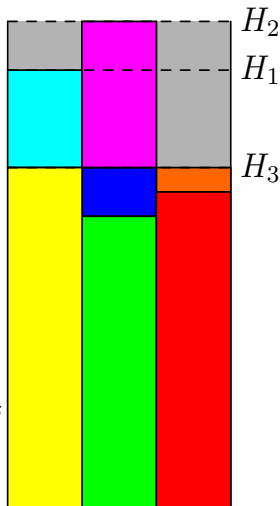
## Définition des problèmes de rangement

- $m$  boîtes  $(B_1, B_2, \dots, B_m)$  de hauteur  $H$
- $n$  objets  $(O_1, O_2, \dots, O_n)$  de hauteur  $h_j$

**question :**

peut-on ranger les  $n$  objets dans  
les  $m$  boîtes de hauteur  $H$  ?

- hauteur dans une boîte :  $H_i = \sum_{O_j \in B_i} h_j$
- ordre indifférent au sein d'une boîte




# Les classiques de la littérature


## Les classiques de la littérature

- Minimiser la hauteur des boîtes

## Les classiques de la littérature

- Minimiser la hauteur des boîtes
  - ordonnancement multiprocesseur ( $P||C_{\max}$ )
    -  Błażewicz *et al.* 96 ; Chen *et al.* 98

## Les classiques de la littérature

- Minimiser la hauteur des boîtes
  - ordonnancement multiprocesseur ( $P||C_{\max}$ )
    -  Błażewicz *et al.* 96 ; Chen *et al.* 98
- Minimiser le nombre de boîtes

## Les classiques de la littérature

- Minimiser la hauteur des boîtes
  - ordonnancement multiprocesseur ( $P||C_{\max}$ )
    - 📖 Błażewicz *et al.* 96 ; Chen *et al.* 98
- Minimiser le nombre de boîtes
  - le *bin-packing*
    - 📖 Coffman *et al.* 97



## Les classiques de la littérature

- Minimiser la hauteur des boîtes
  - ordonnancement multiprocesseur ( $P||C_{\max}$ )
    - 📖 Błażewicz *et al.* 96 ; Chen *et al.* 98
- Minimiser le nombre de boîtes
  - le *bin-packing*
    - 📖 Coffman *et al.* 97
- Maximiser le nombre d'objets rangés

## Les classiques de la littérature

- Minimiser la hauteur des boîtes
  - ordonnancement multiprocesseur ( $P||C_{\max}$ )
    - 📖 Błażewicz *et al.* 96 ; Chen *et al.* 98
- Minimiser le nombre de boîtes
  - le *bin-packing*
    - 📖 Coffman *et al.* 97
- Maximiser le nombre d'objets rangés
  - le *bin-packing* de cardinalité maximum
    - 📖 Coffman *et al.* 78 ; Labbé *et al.* 03

## Les classiques de la littérature

- Minimiser la hauteur des boîtes
  - ordonnancement multiprocesseur ( $P||C_{\max}$ )
    - 📖 Błażewicz *et al.* 96 ; Chen *et al.* 98
- Minimiser le nombre de boîtes
  - le *bin-packing*
    - 📖 Coffman *et al.* 97
- Maximiser le nombre d'objets rangés
  - le *bin-packing* de cardinalité maximum
    - 📖 Coffman *et al.* 78 ; Labbé *et al.* 03
  - le sac-à-dos
    - 📖 Kellerer *et al.* 04

# Expertises et contre-expertises

## Expertises et contre-expertises

- besoin d'évaluations multiples  
(sports artistiques, publications scientifiques, *etc.* )

## Expertises et contre-expertises

- besoin d'évaluations multiples  
(sports artistiques, publications scientifiques, *etc.* )



- chaque document doit être analysé par **plusieurs** experts

## Expertises et contre-expertises

- besoin d'évaluations multiples  
(sports artistiques, publications scientifiques, *etc.* )



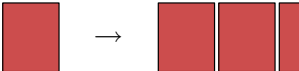
- chaque document doit être analysé par **plusieurs** experts
- un objet  $O_j$  donne  $s_j$  copies



## Expertises et contre-expertises

- besoin d'évaluations multiples  
(sports artistiques, publications scientifiques, *etc.* )



- chaque document doit être analysé par **plusieurs** experts
- un objet  $O_j$  donne  $s_j$  copies 
- chaque copie doit être rangée dans une boîte **différente**



# Classification des rangements d'objets multiboîtes

## Classification des rangements d'objets multiboîtes

$$\alpha|\beta|\gamma$$

## Classification des rangements d'objets multiboîtes

$$\alpha|\beta|\gamma$$

$\alpha$  (les boîtes)

## Classification des rangements d'objets multiboîtes

$$\alpha|\beta|\gamma$$

$\alpha$  (les boîtes)

- $B$  : nombre arbitraire de boîtes identiques

## Classification des rangements d'objets multiboîtes

$$\alpha|\beta|\gamma$$

$\alpha$  (les boîtes)

- $B$  : nombre arbitraire de boîtes identiques
- $Bm$  :  $m$  boîtes identiques

## Classification des rangements d'objets multiboîtes

$$\alpha|\beta|\gamma$$

$\beta$  (les objets)

## Classification des rangements d'objets multiboîtes

$$\alpha|\beta|\gamma$$

$\beta$  (les objets)

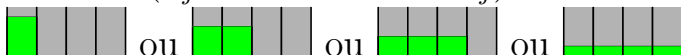
- $size_j$  : objets avec un nombre de copies fixé ( $s_j$  copies de l'objets  $O_j$ )

## Classification des rangements d'objets multiboîtes

$$\alpha|\beta|\gamma$$

$\beta$  (les objets)

- $size_j$  : objets avec un nombre de copies fixé ( $s_j$  copies de l'objets  $O_j$ )
- $any_j$  : objets avec un nombre de copies variable ( $h_j$  est fonction de  $s_j$ )





## Classification des rangements d'objets multiboîtes

$$\alpha|\beta|\gamma$$

$\beta$  (les objets)

- $size_j$  : objets avec un nombre de copies fixé ( $s_j$  copies de l'objets  $O_j$ )
- $any_j$  : objets avec un nombre de copies variable ( $h_j$  est fonction de  $s_j$ )
- $set_j$  : objets ne pouvant aller que dans certains ensembles de boîtes (à préciser)

## Classification des rangements d'objets multiboîtes

$$\alpha|\beta|\gamma$$

$\gamma$  (l'objectif)

## Classification des rangements d'objets multiboîtes

$$\alpha|\beta|\gamma$$

$\gamma$  (l'objectif)

- $H_{\max}$  : minimiser la hauteur des boîtes

## Classification des rangements d'objets multiboîtes

$$\alpha|\beta|\gamma$$

$\gamma$  (l'objectif)

- $H_{\max}$  : minimiser la hauteur des boîtes
- $m$  : minimiser le nombre de boîtes

## Classification des rangements d'objets multiboîtes

$$\alpha|\beta|\gamma$$

$\gamma$  (l'objectif)

- $H_{\max}$  : minimiser la hauteur des boîtes
- $m$  : minimiser le nombre de boîtes
- $N$  : maximiser le nombre d'objets rangés dans  $m$  boîtes de hauteur  $H$

## Différents problèmes d'expertises

## Différents problèmes d'expertises

- $m$  experts

## Différents problèmes d'expertises

- $m$  experts

$\Rightarrow Bm$



## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses

## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses  $\Rightarrow size_j$

## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses  
et de même durée d'analyse  $\Rightarrow size_j$

## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses  $\Rightarrow size_j$   
et de même durée d'analyse  $\Rightarrow h_j = 1$

## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses  $\Rightarrow size_j$   
et de même durée d'analyse  $\Rightarrow h_j = 1$
- finir le plus vite possible

## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses  $\Rightarrow size_j$   
et de même durée d'analyse  $\Rightarrow h_j = 1$
- finir le plus vite possible  $\Rightarrow H_{\max}$

## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
  - $n$  documents d'importances diverses  $\Rightarrow size_j$   
 et de même durée d'analyse  $\Rightarrow h_j = 1$
  - finir le plus vite possible  $\Rightarrow H_{\max}$
- $\Rightarrow Bm | size_j, h_j = 1 | H_{\max}$

## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses  $\Rightarrow size_j$   
 et de même durée d'analyse  $\Rightarrow h_j = 1$
- finir le plus vite possible  $\Rightarrow H_{\max}$

$$\Rightarrow Bm | size_j, h_j = 1 | H_{\max}$$

- des experts



## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses  $\Rightarrow size_j$   
 et de même durée d'analyse  $\Rightarrow h_j = 1$
- finir le plus vite possible  $\Rightarrow H_{\max}$   
 $\Rightarrow Bm|size_j, h_j = 1|H_{\max}$
  
- des experts  $\Rightarrow B$

## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses  $\Rightarrow size_j$   
 et de même durée d'analyse  $\Rightarrow h_j = 1$
- finir le plus vite possible  $\Rightarrow H_{\max}$

$$\Rightarrow Bm | size_j, h_j = 1 | H_{\max}$$

- des experts  $\Rightarrow B$
- $n$  documents partagés

## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses  $\Rightarrow size_j$   
 et de même durée d'analyse  $\Rightarrow h_j = 1$
- finir le plus vite possible  $\Rightarrow H_{\max}$

$$\Rightarrow Bm | size_j, h_j = 1 | H_{\max}$$

- des experts  $\Rightarrow B$
- $n$  documents partagés  $\Rightarrow any_j$

## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses  $\Rightarrow size_j$   
 et de même durée d'analyse  $\Rightarrow h_j = 1$
- finir le plus vite possible  $\Rightarrow H_{\max}$

$$\Rightarrow Bm | size_j, h_j = 1 | H_{\max}$$

- des experts  $\Rightarrow B$
- $n$  documents partagés  $\Rightarrow any_j$
- tenir les délais avec le minimum de personnel

## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses  $\Rightarrow size_j$   
et de même durée d'analyse  $\Rightarrow h_j = 1$
- finir le plus vite possible  $\Rightarrow H_{\max}$

$$\Rightarrow Bm | size_j, h_j = 1 | H_{\max}$$

- des experts  $\Rightarrow B$
- $n$  documents partagés  $\Rightarrow any_j$
- tenir les délais avec le minimum de personnel  $\Rightarrow m$

## Différents problèmes d'expertises

- $m$  experts  $\Rightarrow Bm$
- $n$  documents d'importances diverses  $\Rightarrow size_j$   
 et de même durée d'analyse  $\Rightarrow h_j = 1$
- finir le plus vite possible  $\Rightarrow H_{\max}$

$$\Rightarrow Bm | size_j, h_j = 1 | H_{\max}$$

- des experts  $\Rightarrow B$
- $n$  documents partagés  $\Rightarrow any_j$
- tenir les délais avec le minimum de personnel  $\Rightarrow m$

$$\Rightarrow B | any_j | m$$

# Une vague notion de complexité

## Une vague notion de complexité

- problèmes décidables : solubles avec un ordinateur



## Une vague notion de complexité

- problèmes décidables : solubles avec un ordinateur
- les problèmes peuvent être

## Une vague notion de complexité

- problèmes décidables : solubles avec un ordinateur
- les problèmes peuvent être
  - faciles à résoudre (polynomiaux)

## Une vague notion de complexité

- problèmes décidables : solubles avec un ordinateur
- les problèmes peuvent être
  - faciles à résoudre (polynomiaux)
  - difficiles à résoudre, faciles à vérifier ( $\mathcal{NP}$ -complets)

## Une vague notion de complexité

- problèmes décidables : solubles avec un ordinateur
- les problèmes peuvent être
  - faciles à résoudre (polynomiaux)
  - difficiles à résoudre, faciles à vérifier ( $\mathcal{NP}$ -complets)

|       | taille 10    | taille 30    | taille 60    |
|-------|--------------|--------------|--------------|
| $n$   | 0,00001 sec. | 0,00003 sec. | 0,00006 sec. |
| $n^2$ | 0,0001 sec.  | 0,0009 sec.  | 0,0036 sec.  |
| $n^5$ | 0,1 sec.     | 24,3 sec.    | 13,0 min.    |
| $2^n$ | 0,001 sec.   | 6,5 ans      | 366 siècles  |

## Une vague notion de complexité

- problèmes décidables : solubles avec un ordinateur
- les problèmes peuvent être
  - faciles à résoudre (polynomiaux)
  - difficiles à résoudre, faciles à vérifier ( $\mathcal{NP}$ -complets)

|       | vitesse 1 | vitesse 100  | vitesse 1000 |
|-------|-----------|--------------|--------------|
| $n$   | $N_1$     | $100N_1$     | $1000N_1$    |
| $n^2$ | $N_2$     | $10N_2$      | $31,6N_2$    |
| $n^5$ | $N_3$     | $2,5N_3$     | $3,98N_3$    |
| $2^n$ | $N_4$     | $N_4 + 6,64$ | $N_4 + 9,97$ |

# Complexité des problèmes multiboîtes

## Complexité des problèmes multiboîtes

- $Bm|size_j|\cdot$  est  $\mathcal{NP}$ -complet (au sens ordinaire)

## Complexité des problèmes multiboîtes

- $Bm|size_j|\cdot$  est  $\mathcal{NP}$ -complet (au sens ordinaire)
- $B|size_j|\cdot$  est au moins  $\mathcal{NP}$ -difficile (au sens fort)



## Complexité des problèmes multiboîtes

- $Bm|size_j| \cdot$  est  $\mathcal{NP}$ -complet (au sens ordinaire)
- $B|size_j| \cdot$  est au moins  $\mathcal{NP}$ -difficile (au sens fort)
  
- $Bm|any_j| \cdot$  est  $\mathcal{NP}$ -complet (au sens ordinaire)

## Complexité des problèmes multiboîtes

- $Bm|size_j| \cdot$  est  $\mathcal{NP}$ -complet (au sens ordinaire)
- $B|size_j| \cdot$  est au moins  $\mathcal{NP}$ -difficile (au sens fort)
  
- $Bm|any_j| \cdot$  est  $\mathcal{NP}$ -complet (au sens ordinaire)
- $B|any_j| \cdot$  est  $\mathcal{NP}$ -complet (au sens fort)

## Complexité des problèmes multiboîtes

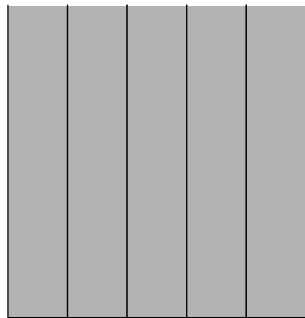
- $Bm|size_j| \cdot$  est  $\mathcal{NP}$ -complet (au sens ordinaire)
- $B|size_j| \cdot$  est au moins  $\mathcal{NP}$ -difficile (au sens fort)
  
- $Bm|any_j| \cdot$  est  $\mathcal{NP}$ -complet (au sens ordinaire)
- $B|any_j| \cdot$  est  $\mathcal{NP}$ -complet (au sens fort)
  
- $B|size_j, h_j = 1| \cdot$  est polynomial

*Premier mouvement*

A musical staff with five lines, tilted upwards from left to right. It contains several musical notes, including quarter notes and eighth notes, some with stems pointing up and some pointing down. The notes are rendered in a light gray color.

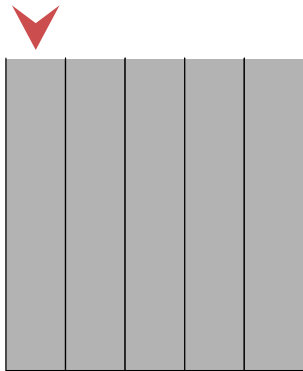
*Résolutions exactes  
de quelques cas*

A musical staff with five lines, tilted upwards from left to right. It contains several musical notes, including quarter notes and eighth notes, some with stems pointing up and some pointing down. The notes are rendered in a light gray color.

Résolution de  $B|size_j, h_j = 1|H_{\max}$ 

## Algorithme 1

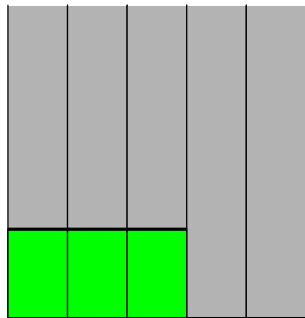
- ①  $i = 1$
- ② pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

Résolution de  $B|size_j, h_j = 1|H_{\max}$ 

## Algorithme 1

- ①  $i = 1$
- ② pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

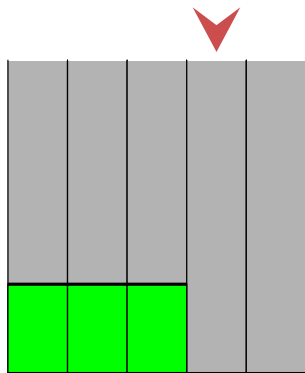
Résolution de  $B|size_j, h_j = 1|H_{\max}$



### Algorithme 1

- ①  $i = 1$
- ② pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

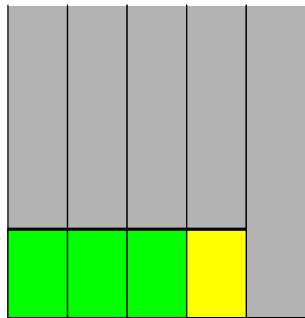
Résolution de  $B|size_j, h_j = 1|H_{\max}$



### Algorithme 1

- ①  $i = 1$
- ② pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

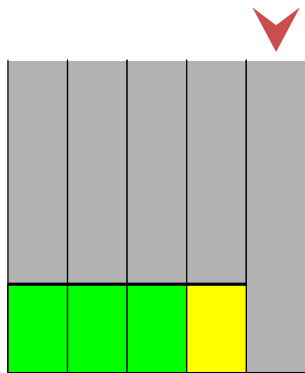


Résolution de  $B|size_j, h_j = 1|H_{\max}$ 

## Algorithme 1

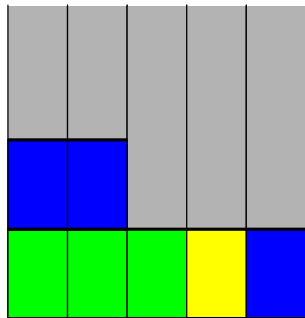
- ①  $i = 1$
- ② pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

Résolution de  $B|size_j, h_j = 1|H_{\max}$



### Algorithme 1

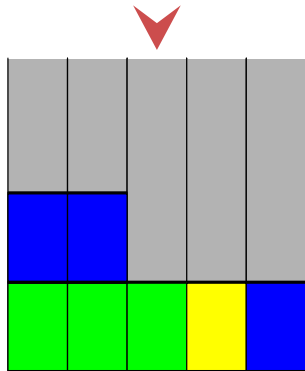
- ①  $i = 1$
- ② pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

Résolution de  $B|size_j, h_j = 1|H_{\max}$ 

## Algorithme 1

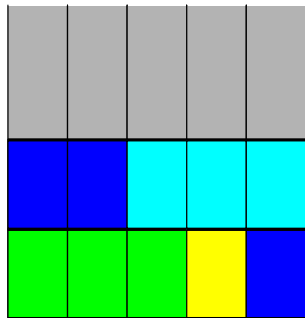
- ①  $i = 1$
- ② pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

Résolution de  $B|size_j, h_j = 1|H_{\max}$



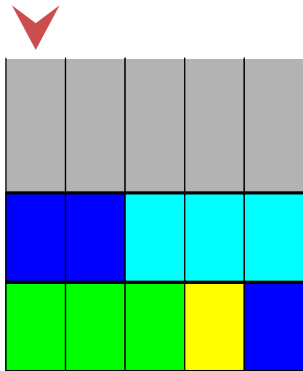
### Algorithme 1

- ①  $i = 1$
- ② pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

Résolution de  $B|size_j, h_j = 1|H_{\max}$ 

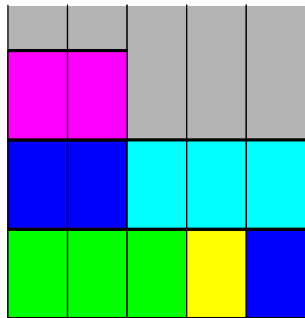
## Algorithme 1

- ①  $i = 1$
- ② pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

Résolution de  $B|size_j, h_j = 1|H_{\max}$ 

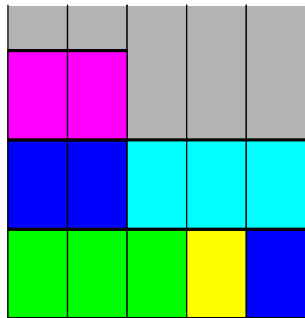
## Algorithme 1

- ①  $i = 1$
- ② pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

Résolution de  $B|size_j, h_j = 1|H_{\max}$ 

## Algorithme 1

- ①  $i = 1$
- ② pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

Résolution de  $B|size_j, h_j = 1|H_{\max}$ 

## Algorithme 1

- 1  $i = 1$
- 2 pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

## Propriétés

- algorithme optimal
- complexité  $\mathcal{O}(n)$



Résolution de  $B|size_j, h_j = 1|$ .

Résolution de  $B|size_j, h_j = 1|$ .

- $B|size_j, h_j = 1|H_{\max}$

Résolution de  $B|size_j, h_j = 1|$ .

- $B|size_j, h_j = 1|H_{\max}$ 
  - utiliser l'algorithme 1
  - valeur optimale :  $H_{\max}^* = \left\lceil \frac{\sum_j s_j}{m} \right\rceil$

Résolution de  $B|size_j, h_j = 1|$ .

- $B|size_j, h_j = 1|H_{\max}$ 
  - utiliser l'algorithme 1
  - valeur optimale :  $H_{\max}^* = \left\lceil \frac{\sum_j s_j}{m} \right\rceil$
- $B|size_j, h_j = 1|m$

Résolution de  $B|size_j, h_j = 1|$ .

- $B|size_j, h_j = 1|H_{\max}$ 
  - utiliser l'algorithme 1
  - valeur optimale :  $H_{\max}^* = \left\lceil \frac{\sum_j s_j}{m} \right\rceil$
  
- $B|size_j, h_j = 1|m$ 
  - calculer la valeur optimale :  $m^* = \max \left\{ s_{\max}; \left\lceil \frac{\sum_j s_j}{H} \right\rceil \right\}$
  - puis utiliser l'algorithme 1 avec  $m^*$

Résolution de  $B|size_j, h_j = 1|$ .

- $B|size_j, h_j = 1|H_{\max}$ 
  - utiliser l'algorithme 1
  - valeur optimale :  $H_{\max}^* = \left\lceil \frac{\sum_j s_j}{m} \right\rceil$
  
- $B|size_j, h_j = 1|m$ 
  - calculer la valeur optimale :  $m^* = \max \left\{ s_{\max}; \left\lceil \frac{\sum_j s_j}{H} \right\rceil \right\}$
  - puis utiliser l'algorithme 1 avec  $m^*$
  
- $B|size_j, h_j = 1|N$

Résolution de  $B|size_j, h_j = 1|$ .

- $B|size_j, h_j = 1|H_{\max}$ 
  - utiliser l'algorithme 1
  - valeur optimale :  $H_{\max}^* = \left\lceil \frac{\sum_j s_j}{m} \right\rceil$
  
- $B|size_j, h_j = 1|m$ 
  - calculer la valeur optimale :  $m^* = \max \left\{ s_{\max}; \left\lceil \frac{\sum_j s_j}{H} \right\rceil \right\}$
  - puis utiliser l'algorithme 1 avec  $m^*$
  
- $B|size_j, h_j = 1|N$ 
  - trier les objets :  $s_1 \leq s_2 \leq \dots \leq s_n$
  - puis utiliser l'algorithme 1 avec  $m$  et  $H$  donnés

Résolution de  $B2|size_j|H_{\max}$



Résolution de  $B2|size_j|H_{\max}$ 

- $P(j, H_1, H_2)$  : valeur optimale

Résolution de  $B2|size_j|H_{\max}$ 

- $P(j, H_1, H_2)$  : valeur optimale
  - pour ranger  $O_j, O_{j+1}, \dots, O_n$

Résolution de  $B2|size_j|H_{\max}$ 

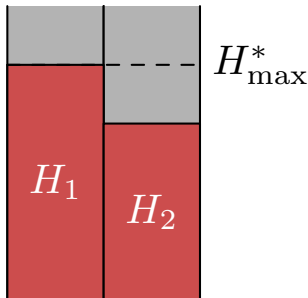
- $P(j, H_1, H_2)$  : valeur optimale
  - pour ranger  $O_j, O_{j+1}, \dots, O_n$
  - sachant qu'il y a déjà les hauteurs  $H_1$  et  $H_2$  dans les boîtes

Résolution de  $B2|size_j|H_{\max}$ 

- $P(j, H_1, H_2)$  : valeur optimale
  - pour ranger  $O_j, O_{j+1}, \dots, O_n$
  - sachant qu'il y a déjà les hauteurs  $H_1$  et  $H_2$  dans les boîtes
- on cherche  $P(1, 0, 0)$

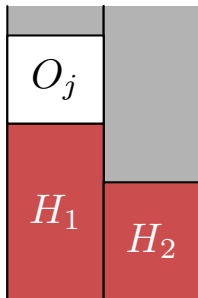
Résolution de  $B2|size_j|H_{\max}$ 

- $P(j, H_1, H_2)$  : valeur optimale
  - pour ranger  $O_j, O_{j+1}, \dots, O_n$
  - sachant qu'il y a déjà les hauteurs  $H_1$  et  $H_2$  dans les boîtes
- on cherche  $P(1, 0, 0)$
- si  $j > n$  alors  $P(j, H_1, H_2) = \max(H_1, H_2)$



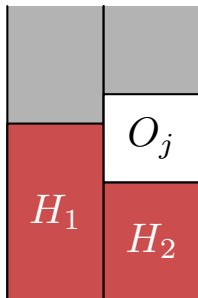
Résolution de  $B2|size_j|H_{\max}$ 

- $P(j, H_1, H_2)$  : valeur optimale
  - pour ranger  $O_j, O_{j+1}, \dots, O_n$
  - sachant qu'il y a déjà les hauteurs  $H_1$  et  $H_2$  dans les boîtes
- on cherche  $P(1, 0, 0)$
- si  $j > n$  alors  $P(j, H_1, H_2) = \max(H_1, H_2)$
- si  $j \leq n$  alors  $P(j, H_1, H_2) = \min(P(j+1, H_1 + h_j, H_2), P(j+1, H_1, H_2 + h_j))$



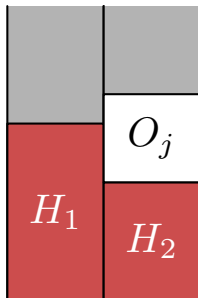
Résolution de  $B2|size_j|H_{\max}$ 

- $P(j, H_1, H_2)$  : valeur optimale
  - pour ranger  $O_j, O_{j+1}, \dots, O_n$
  - sachant qu'il y a déjà les hauteurs  $H_1$  et  $H_2$  dans les boîtes
- on cherche  $P(1, 0, 0)$
- si  $j > n$  alors  $P(j, H_1, H_2) = \max(H_1, H_2)$
- si  $j \leq n$  alors  $P(j, H_1, H_2) = \min(P(j+1, H_1 + h_j, H_2), P(j+1, H_1, H_2 + h_j))$



Résolution de  $B2|size_j|H_{\max}$ 

- $P(j, H_1, H_2)$  : valeur optimale
  - pour ranger  $O_j, O_{j+1}, \dots, O_n$
  - sachant qu'il y a déjà les hauteurs  $H_1$  et  $H_2$  dans les boîtes
- on cherche  $P(1, 0, 0)$
- si  $j > n$  alors  $P(j, H_1, H_2) = \max(H_1, H_2)$
- si  $j \leq n$  alors  $P(j, H_1, H_2) = \min(P(j+1, H_1 + h_j, H_2), P(j+1, H_1, H_2 + h_j))$



## Propriétés

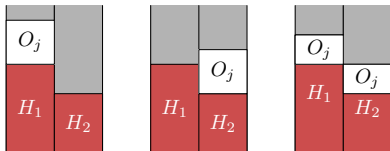
- algorithme optimal
- complexité  $\mathcal{O}(nA^2)$   
avec  $A = \sum_j h_j$



Extension à  $Bm|any_j|$ .

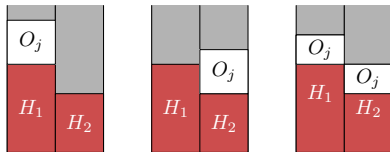
Extension à  $Bm|any_j|$ .

- extension au cas  $any_j$



Extension à  $Bm|any_j|$ .

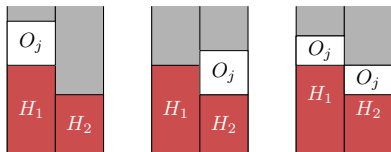
- extension au cas  $any_j$



- extension à un nombre de boîtes quelconque (fixé)

Extension à  $Bm|any_j|$ .

- extension au cas  $any_j$



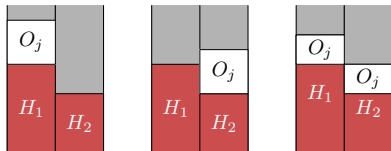
- extension à un nombre de boîtes quelconque (fixé)

## Théorème

il existe un algorithme optimal et pseudo-polynomial en temps  $\mathcal{O}(nA^m)$  pour  $Bm|any_j|H_{\max}$

Extension à  $Bm|any_j|$ .

- extension au cas  $any_j$



- extension à un nombre de boîtes quelconque (fixé)

## Théorème

il existe un algorithme optimal et pseudo-polynomial en temps  $\mathcal{O}(nA^m)$  pour  $Bm|any_j|H_{\max}$

- adaptation possible à l'objectif  $N$

# *Deuxième mouvement*

The background of the slide features two musical staves with notes, rendered in a light gray, semi-transparent style. The top staff shows a sequence of notes, and the bottom staff shows a similar sequence, both appearing to be part of a larger musical piece.

*Résolution approchée  
de  $B|size_j|H_{\max}$*

# Les principes du « redimensionnement »

## Les principes du « redimensionnement »

- **problème** : il y a trop de hauteurs différentes  
(on doit considérer toutes les hauteurs jusqu'à  $A = \sum h_j$ )



## Les principes du « redimensionnement »

- **problème** : il y a trop de hauteurs différentes  
(on doit considérer toutes les hauteurs jusqu'à  $A = \sum h_j$ )
- **solution** : réduire le nombre de hauteurs pour que l'algorithme soit polynomial

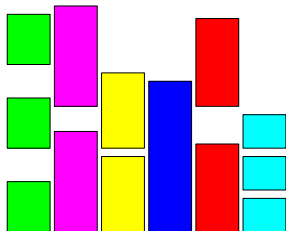
## Les principes du « redimensionnement »

- **problème** : il y a trop de hauteurs différentes  
(on doit considérer toutes les hauteurs jusqu'à  $A = \sum h_j$ )
- **solution** : réduire le nombre de hauteurs pour que l'algorithme soit polynomial
- **limites** : on perd l'optimalité (mais on garde un bon contrôle)

## Les principes du « redimensionnement »

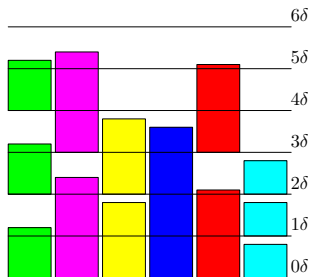
- **problème** : il y a trop de hauteurs différentes  
(on doit considérer toutes les hauteurs jusqu'à  $A = \sum h_j$ )
- **solution** : réduire le nombre de hauteurs pour que l'algorithme soit polynomial
- **limites** : on perd l'optimalité (mais on garde un bon contrôle)
  
- adapté aux problèmes numériques pseudo-polynomiaux  
(ici : nombres = hauteurs)

 Ausiello *et al.* 99

Un schéma approximation pour  $Bm|size_j|H_{\max}$ 

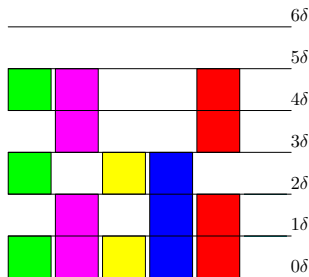
## Algorithme FPTAS

- ① calculer  $\delta = \varepsilon \frac{h_{\max}}{n}$
- ② réduire les hauteurs :  $h'_j = \left\lfloor \frac{h_j}{\delta} \right\rfloor$
- ③ résoudre avec l'algorithme pseudo-polynomial
- ④ redonner aux objets leurs hauteurs réelles

Un schéma approximation pour  $Bm|size_j|H_{\max}$ 

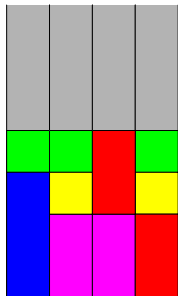
## Algorithme FPTAS

- ① calculer  $\delta = \varepsilon \frac{h_{\max}}{n}$
- ② réduire les hauteurs :  $h'_j = \left\lfloor \frac{h_j}{\delta} \right\rfloor$
- ③ résoudre avec l'algorithme pseudo-polynomial
- ④ redonner aux objets leurs hauteurs réelles

Un schéma approximation pour  $Bm|size_j|H_{\max}$ 

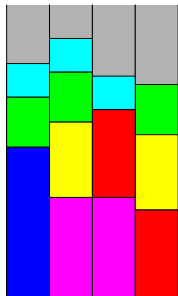
## Algorithme FPTAS

- 1 calculer  $\delta = \varepsilon \frac{h_{\max}}{n}$
- 2 réduire les hauteurs :  $h'_j = \left\lfloor \frac{h_j}{\delta} \right\rfloor$
- 3 résoudre avec l'algorithme pseudo-polynomial
- 4 redonner aux objets leurs hauteurs réelles

Un schéma approximation pour  $Bm|size_j|H_{\max}$ 

## Algorithme FPTAS

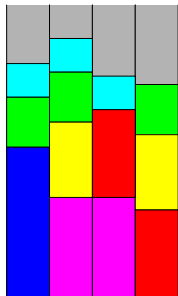
- ① calculer  $\delta = \varepsilon \frac{h_{\max}}{n}$
- ② réduire les hauteurs :  $h'_j = \left\lfloor \frac{h_j}{\delta} \right\rfloor$
- ③ résoudre avec l'algorithme pseudo-polynomial
- ④ redonner aux objets leurs hauteurs réelles

Un schéma approximation pour  $Bm|size_j|H_{\max}$ 

## Algorithme FPTAS

- ① calculer  $\delta = \varepsilon \frac{h_{\max}}{n}$
- ② réduire les hauteurs :  $h'_j = \left\lfloor \frac{h_j}{\delta} \right\rfloor$
- ③ résoudre avec l'algorithme pseudo-polynomial
- ④ redonner aux objets leurs hauteurs réelles



Un schéma approximation pour  $Bm|size_j|H_{\max}$ 

## Algorithme FPTAS

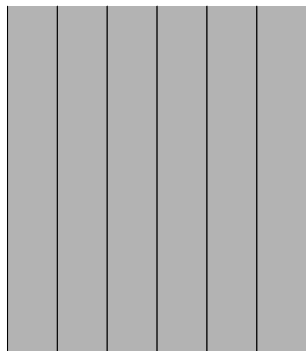
- ① calculer  $\delta = \varepsilon \frac{h_{\max}}{n}$
- ② réduire les hauteurs :  $h'_j = \left\lfloor \frac{h_j}{\delta} \right\rfloor$
- ③ résoudre avec l'algorithme pseudo-polynomial
- ④ redonner aux objets leurs hauteurs réelles

## Propriétés

$(1 + \varepsilon)$ -approximation :

- $H_{\max} \leq (1 + \varepsilon)H_{\max}^*$
- complexité  $\mathcal{O}\left(\frac{n^{2m+1}}{\varepsilon^m}\right)$

# Placement dans les boîtes suivantes décroissant



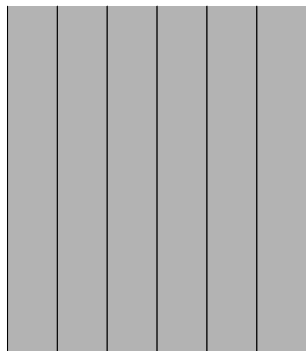
## Algorithme PBSD

- ① trier les objets :  

$$h_1 \geq h_2 \geq \dots \geq h_n$$
- ②  $i = 1$
- ③ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  

$$B_i, \dots, B_{i+size_j-1}$$
  - $i = ((i + size_j - 1) \bmod m) + 1$

## Placement dans les boîtes suivantes décroissant

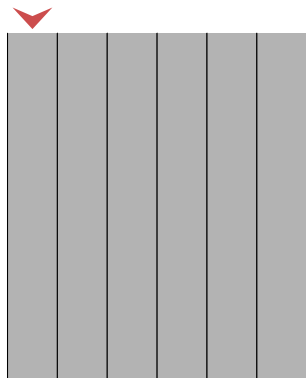


## Algorithme PBSD

- ① trier les objets :  

$$h_1 \geq h_2 \geq \dots \geq h_n$$
- ②  $i = 1$
- ③ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  
 $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

## Placement dans les boîtes suivantes décroissant



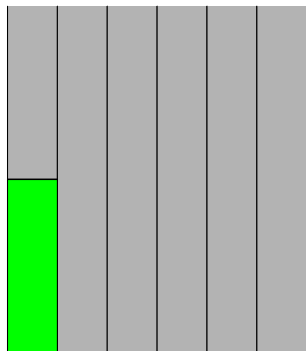
## Algorithme PBSD

- ① trier les objets :  

$$h_1 \geq h_2 \geq \dots \geq h_n$$
- ②  $i = 1$
- ③ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  

$$B_i, \dots, B_{i+size_j-1}$$
  - $i = ((i + size_j - 1) \bmod m) + 1$

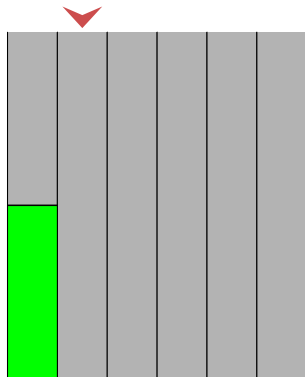
## Placement dans les boîtes suivantes décroissant



## Algorithme PBSD

- ❶ trier les objets :  
 $h_1 \geq h_2 \geq \dots \geq h_n$
- ❷  $i = 1$
- ❸ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  
 $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

## Placement dans les boîtes suivantes décroissant



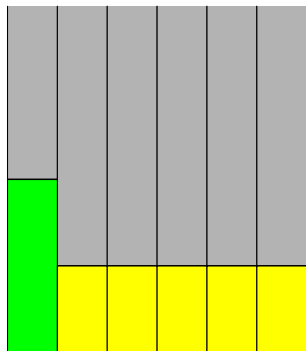
## Algorithme PBSB

- ① trier les objets :  

$$h_1 \geq h_2 \geq \dots \geq h_n$$
- ②  $i = 1$
- ③ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  

$$B_i, \dots, B_{i+size_j-1}$$
  - $i = ((i + size_j - 1) \bmod m) + 1$

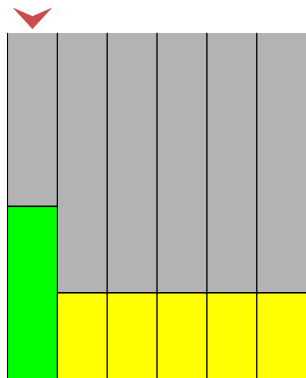
## Placement dans les boîtes suivantes décroissant



## Algorithme PBSB

- ① trier les objets :  
 $h_1 \geq h_2 \geq \dots \geq h_n$
- ②  $i = 1$
- ③ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  
 $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

## Placement dans les boîtes suivantes décroissant



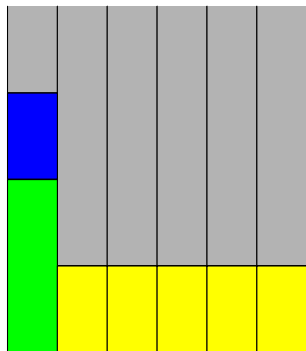
## Algorithme PBSO

- ① trier les objets :  

$$h_1 \geq h_2 \geq \dots \geq h_n$$
- ②  $i = 1$
- ③ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  
 $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$



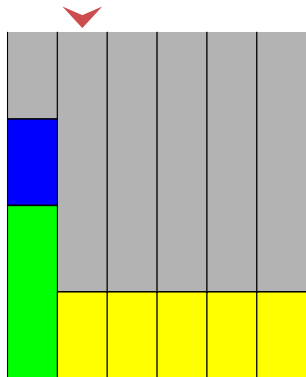
## Placement dans les boîtes suivantes décroissant



## Algorithme PBSD

- ① trier les objets :  
 $h_1 \geq h_2 \geq \dots \geq h_n$
- ②  $i = 1$
- ③ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  
 $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

## Placement dans les boîtes suivantes décroissant

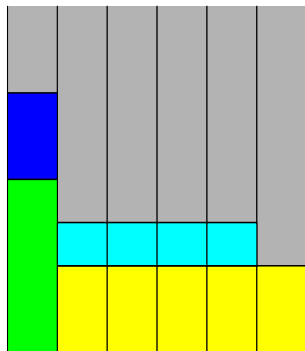


## Algorithme PBSO

- ① trier les objets :  

$$h_1 \geq h_2 \geq \dots \geq h_n$$
- ②  $i = 1$
- ③ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  
 $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

## Placement dans les boîtes suivantes décroissant

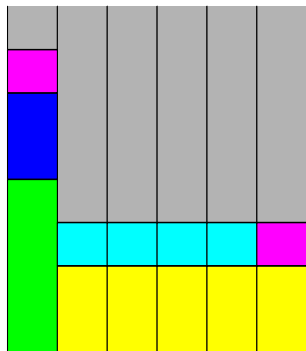


## Algorithme PBSO

- ① trier les objets :  

$$h_1 \geq h_2 \geq \dots \geq h_n$$
- ②  $i = 1$
- ③ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  
 $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

## Placement dans les boîtes suivantes décroissant

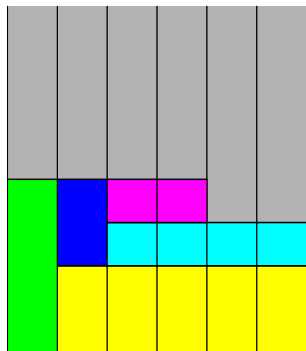


## Algorithme PBSO

- ① trier les objets :  

$$h_1 \geq h_2 \geq \dots \geq h_n$$
- ②  $i = 1$
- ③ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  
 $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

## Placement dans les boîtes suivantes décroissant

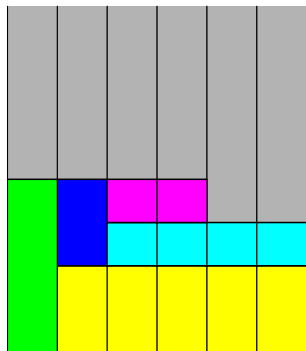


## Algorithme PBSO

- ① trier les objets :  

$$h_1 \geq h_2 \geq \dots \geq h_n$$
- ②  $i = 1$
- ③ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  
 $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

## Placement dans les boîtes suivantes décroissant



## Algorithme PBSO

- ① trier les objets :  

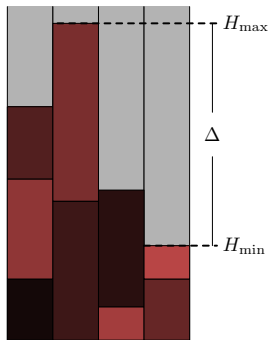
$$h_1 \geq h_2 \geq \dots \geq h_n$$
- ②  $i = 1$
- ③ pour  $j$  de 1 à  $n$  faire
  - placer  $O_j$  dans  
 $B_i, \dots, B_{i+size_j-1}$
  - $i = ((i + size_j - 1) \bmod m) + 1$

## Propriétés

- 2-approximation
- complexité  $\mathcal{O}(n \ln n)$

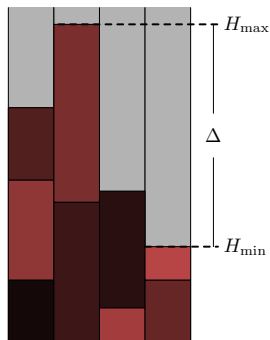
# Garantie pour une solution quelconque

# Garantie pour une solution quelconque



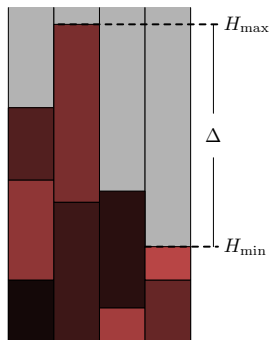


# Garantie pour une solution quelconque



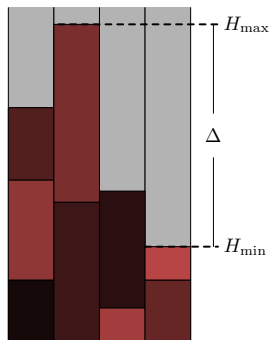
- $mH_{\max}^* \geq \sum_j h_j s_j$

## Garantie pour une solution quelconque



- $mH_{\max}^* \geq \sum_j h_j s_j$
- $\sum_j h_j s_j \leq H_{\max} + (m - 1)(H_{\max} - \Delta)$

## Garantie pour une solution quelconque



- $mH_{\max}^* \geq \sum_j h_j s_j$
- $\sum_j h_j s_j \leq H_{\max} + (m - 1)(H_{\max} - \Delta)$

↓

Propriété

$$H_{\max} \leq H_{\max}^* + \frac{m-1}{m} \Delta$$

# Algorithme « diviser pour régner »

## Algorithme DR

- 1 créer  $n$  solutions partielles
- 2 tant qu'il reste plus d'une solution faire
  - fusionner les solutions deux à deux
  - trier les boîtes au sein des solutions

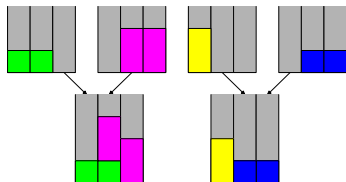
# Algorithme « diviser pour régner »



## Algorithme DR

- 1 créer  $n$  solutions partielles
- 2 tant qu'il reste plus d'une solution faire
  - fusionner les solutions deux à deux
  - trier les boîtes au sein des solutions

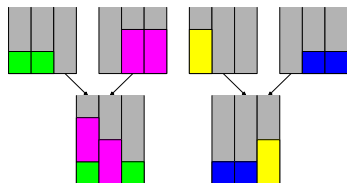
# Algorithme « diviser pour régner »



## Algorithme DR

- ① créer  $n$  solutions partielles
- ② tant qu'il reste plus d'une solution faire
  - fusionner les solutions deux à deux
  - trier les boîtes au sein des solutions

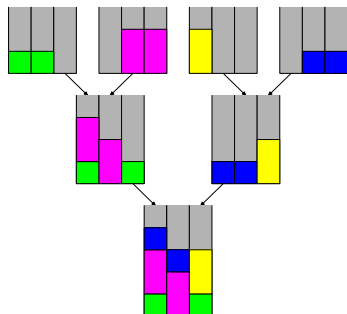
# Algorithme « diviser pour régner »



## Algorithme DR

- ① créer  $n$  solutions partielles
- ② tant qu'il reste plus d'une solution faire
  - fusionner les solutions deux à deux
  - **trier les boîtes au sein des solutions**

# Algorithme « diviser pour régner »

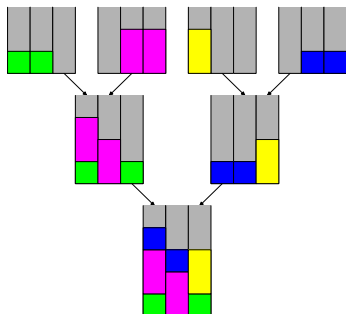


## Algorithme DR

- 1 créer  $n$  solutions partielles
- 2 tant qu'il reste plus d'une solution faire
  - fusionner les solutions deux à deux
  - trier les boîtes au sein des solutions



# Algorithme « diviser pour régner »



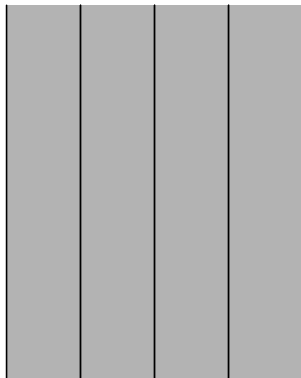
## Algorithme DR

- ① créer  $n$  solutions partielles
- ② tant qu'il reste plus d'une solution faire
  - fusionner les solutions deux à deux
  - trier les boîtes au sein des solutions

## Propriétés

- $H_{\max} \leq H_{\max}^* + \frac{m-1}{m} h_{\max}$
- complexité  $\mathcal{O}(nm \ln m)$

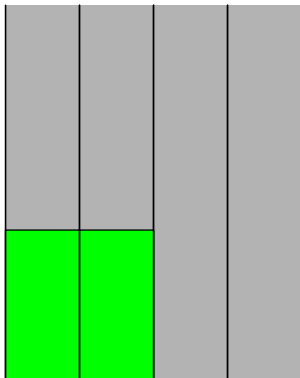
# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - trier les boîtes

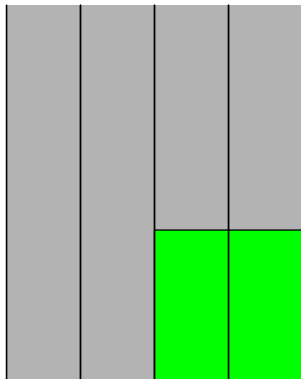
# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - trier les boîtes

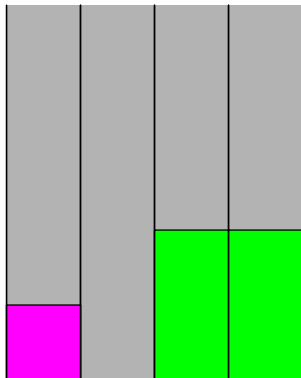
# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - **trier les boîtes**

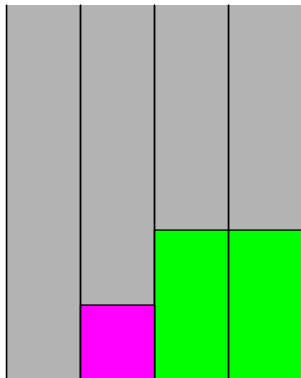
# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - trier les boîtes

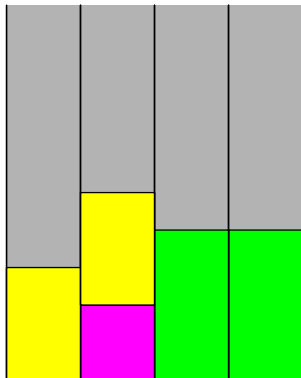
# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - **trier les boîtes**

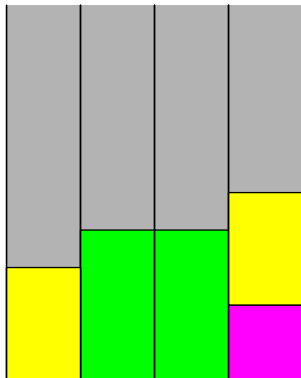
# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - trier les boîtes

# Algorithme de meilleur placement

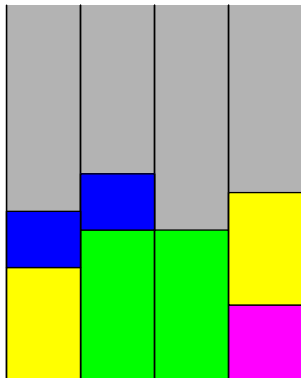


## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - **trier les boîtes**



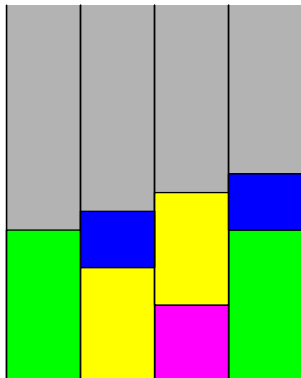
# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - trier les boîtes

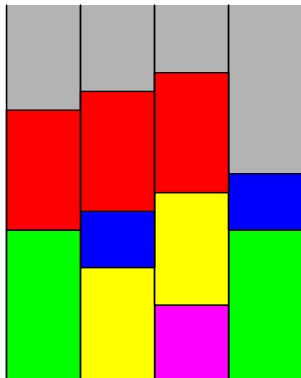
# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - **trier les boîtes**

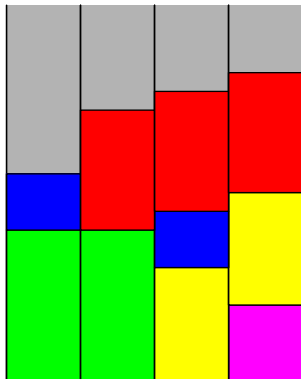
# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - trier les boîtes

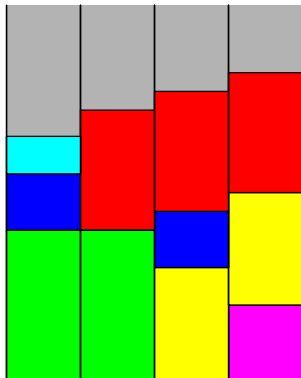
# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - **trier les boîtes**

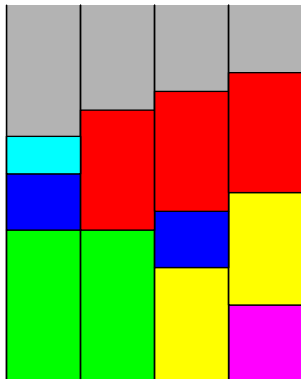
# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - trier les boîtes

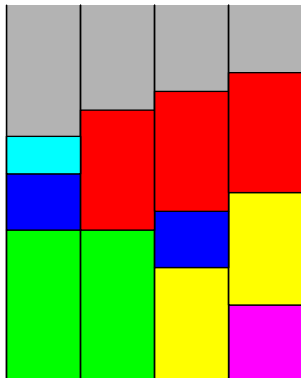
# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - **trier les boîtes**

# Algorithme de meilleur placement



## Algorithme MP

- 1 pour  $j$  de 1 à  $n$  faire
  - ranger  $O_j$  dans  $B_1, B_2, \dots, B_{s_j}$
  - trier les boîtes

## Propriétés

- $H_{\max} \leq H_{\max}^* + \frac{m-1}{m} h_{\max}$
- complexité  $\mathcal{O}(nm)$

## Meilleur placement et pré-affectations

### Algorithme MP avec pré-affectations

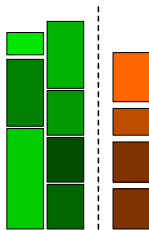
- 1 séparer les objets en deux ensembles  $O_1$  et  $O_2$
- 2 ranger optimalement  $O_1$
- 3 compléter avec  $O_2$  selon MP



# Meilleur placement et pré-affectations

## Algorithme MP avec pré-affectations

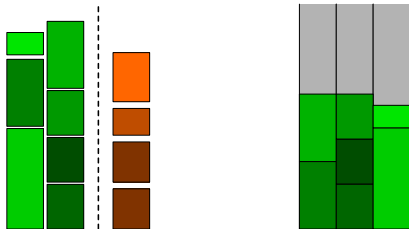
- ① séparer les objets en deux ensembles  $O_1$  et  $O_2$
- ② ranger optimalement  $O_1$
- ③ compléter avec  $O_2$  selon MP



# Meilleur placement et pré-affectations

## Algorithme MP avec pré-affectations

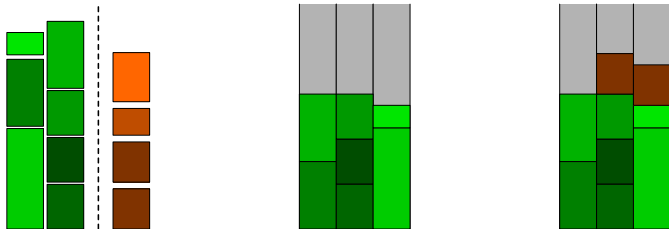
- ① séparer les objets en deux ensembles  $O_1$  et  $O_2$
- ② ranger optimalement  $O_1$
- ③ compléter avec  $O_2$  selon MP



# Meilleur placement et pré-affectations

## Algorithme MP avec pré-affectations

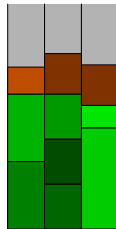
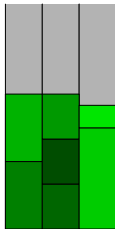
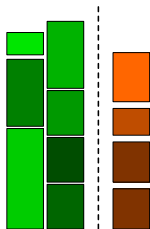
- ① séparer les objets en deux ensembles  $O_1$  et  $O_2$
- ② ranger optimalement  $O_1$
- ③ compléter avec  $O_2$  selon MP



# Meilleur placement et pré-affectations

## Algorithme MP avec pré-affectations

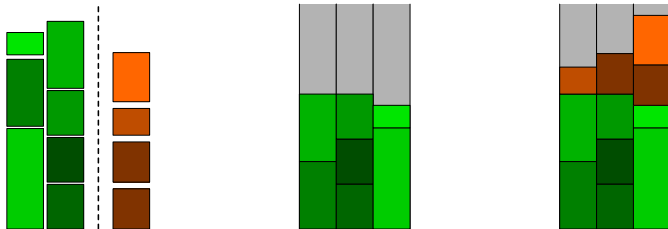
- ① séparer les objets en deux ensembles  $O_1$  et  $O_2$
- ② ranger optimalement  $O_1$
- ③ compléter avec  $O_2$  selon MP



# Meilleur placement et pré-affectations

## Algorithme MP avec pré-affectations

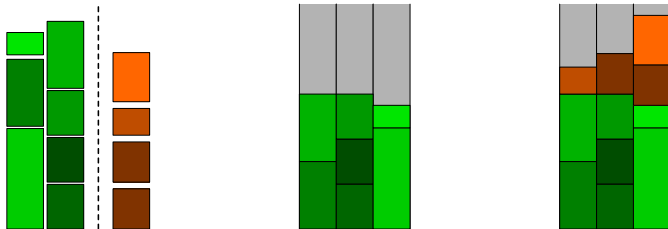
- ① séparer les objets en deux ensembles  $O_1$  et  $O_2$
- ② ranger optimalement  $O_1$
- ③ compléter avec  $O_2$  selon MP



# Meilleur placement et pré-affectations

## Algorithme MP avec pré-affectations

- ① séparer les objets en deux ensembles  $\mathbf{O}_1$  et  $\mathbf{O}_2$
- ② ranger optimalement  $\mathbf{O}_1$
- ③ compléter avec  $\mathbf{O}_2$  selon MP



## Théorème

$$H_{\max} \leq H_{\max}^* + h_{\max}(\mathbf{O}_2)$$

## Meilleur placement décroissant

### Algorithme MPD

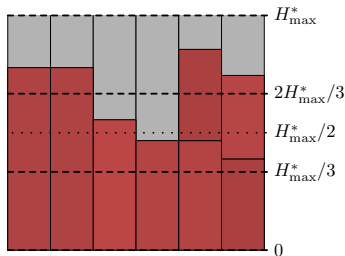
- 1 trier les objets :  $h_1 \geq h_2 \geq \dots \geq h_n$
- 2 utiliser l'algorithme MP

# Meilleur placement décroissant

## Algorithme MPD

- 1 trier les objets :  $h_1 \geq h_2 \geq \dots \geq h_n$
- 2 utiliser l'algorithme MP

- $\mathbf{O}_1 = \{O_j; h_j > H_{\max}^*/3\}$  rangé optimalement par MPD.



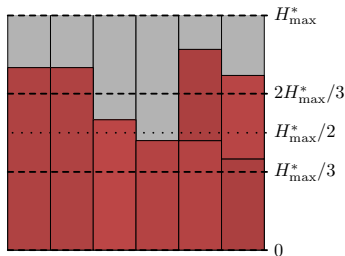


# Meilleur placement décroissant

## Algorithme MPD

- ① trier les objets :  $h_1 \geq h_2 \geq \dots \geq h_n$
- ② utiliser l'algorithme MP

- $\mathbf{O}_1 = \{O_j; h_j > H_{\max}^*/3\}$  rangé optimalement par MPD.
- $\mathbf{O}_2 = \{O_j; h_j \leq H_{\max}^*/3\}$

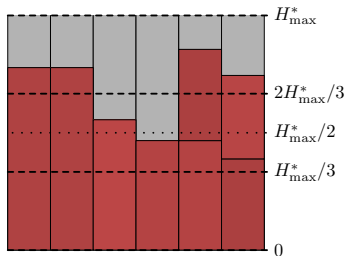


# Meilleur placement décroissant

## Algorithme MPD

- 1 trier les objets :  $h_1 \geq h_2 \geq \dots \geq h_n$
- 2 utiliser l'algorithme MP

- $\mathbf{O}_1 = \{O_j; h_j > H_{\max}^*/3\}$  rangé optimalement par MPD.
- $\mathbf{O}_2 = \{O_j; h_j \leq H_{\max}^*/3\}$



## Propriété

$$H_{\max} \leq \frac{4}{3} H_{\max}^*$$

# Données expérimentales

## Données expérimentales

- données générées aléatoirement
  - nombre d'objets : de 5 à 500
  - nombre de boîtes : de 2 à 20
  - hauteurs des objets : de 1 à 50
  - largeurs des objets : de 1 à  $m$

## Performances des heuristiques

|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |

## Performances des heuristiques

|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |

## Performances des heuristiques

|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |

## Performances des heuristiques

|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |



## Performances des heuristiques

|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |

## Performances des heuristiques

|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |

## Performances des heuristiques

|                     | PBSD        | DR          | MP          | MPD         |
|---------------------|-------------|-------------|-------------|-------------|
| la meilleure (%)    | 67,5        | 78,0        | 70,7        | 86,3        |
| <b>optimale (%)</b> | <b>67,1</b> | <b>68,5</b> | <b>67,4</b> | <b>73,3</b> |
| écart moyen (%)     | 0,7         | 0,5         | 0,6         | 0,3         |
| écart max (%)       | 7,0         | 5,6         | 6,5         | 3,3         |

## Performances des heuristiques

|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |

## Performances des heuristiques

|                      | PBSD       | DR         | MP         | MPD        |
|----------------------|------------|------------|------------|------------|
| la meilleure (%)     | 67,5       | 78,0       | 70,7       | 86,3       |
| optimale (%)         | 67,1       | 68,5       | 67,4       | 73,3       |
| écart moyen (%)      | 0,7        | 0,5        | 0,6        | 0,3        |
| <b>écart max (%)</b> | <b>7,0</b> | <b>5,6</b> | <b>6,5</b> | <b>3,3</b> |

## Performances des heuristiques

|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |

- beaucoup d'instances très faciles

## Performances des heuristiques

|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |

- beaucoup d'instances très faciles
- PBSD efficace seulement sur ces instances

## Performances des heuristiques

|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |

- beaucoup d'instances très faciles
- PBSD efficace seulement sur ces instances
- résolution « en-ligne » : MP est bon



## Performances des heuristiques

|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |

- beaucoup d'instances très faciles
- PBSD efficace seulement sur ces instances
- résolution « en-ligne » : MP est bon
- résolution « hors-ligne » : MPD est nettement dominant

## Performances des heuristiques

|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |

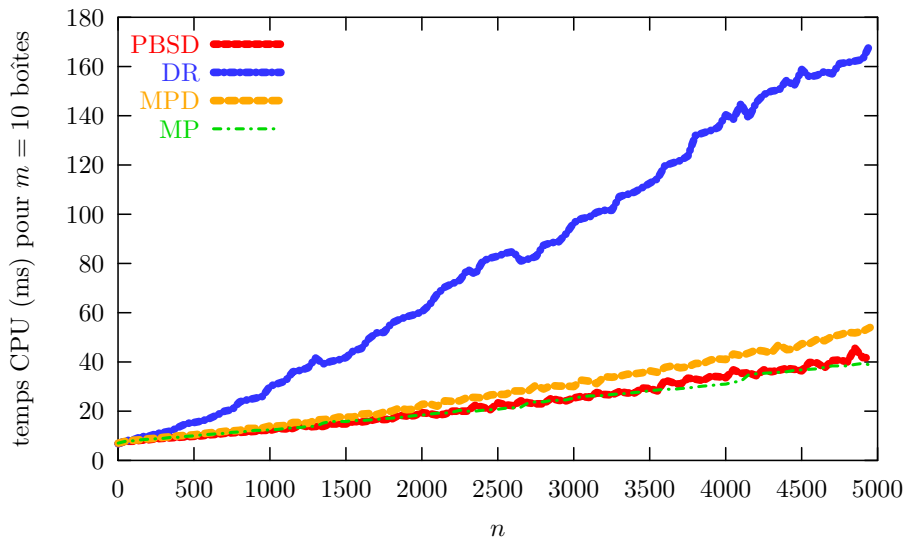
- beaucoup d'instances très faciles
- PBSD efficace seulement sur ces instances
- résolution « en-ligne » : MP est bon
- résolution « hors-ligne » : MPD est nettement dominant
- le potentiel de DR n'est pas complètement exploité

## Performances des heuristiques

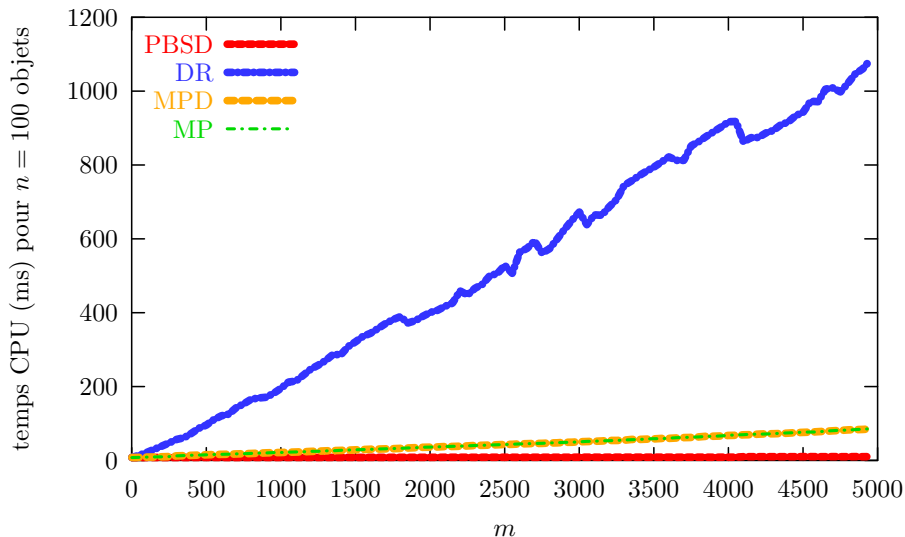
|                  | PBSD | DR   | MP   | MPD  |
|------------------|------|------|------|------|
| la meilleure (%) | 67,5 | 78,0 | 70,7 | 86,3 |
| optimale (%)     | 67,1 | 68,5 | 67,4 | 73,3 |
| écart moyen (%)  | 0,7  | 0,5  | 0,6  | 0,3  |
| écart max (%)    | 7,0  | 5,6  | 6,5  | 3,3  |

- beaucoup d'instances très faciles
- PBSD efficace seulement sur ces instances
- résolution « en-ligne » : MP est bon
- résolution « hors-ligne » : MPD est nettement dominant
- le potentiel de DR n'est pas complètement exploité
- heuristiques assez complémentaires

## Temps de calcul



## Temps de calcul



# Synthèse sur les heuristiques

## Synthèse sur les heuristiques

- heuristiques très rapides et de bonne qualité (théorique et pratique)

## Synthèse sur les heuristiques

- heuristiques très rapides et de bonne qualité (théorique et pratique)
- complémentarité des heuristiques



## Synthèse sur les heuristiques

- heuristiques très rapides et de bonne qualité (théorique et pratique)
- complémentarité des heuristiques
  
- algorithme génétique ayant un très bon comportement (algorithme hybride basé sur l'heuristique MP)

## Synthèse sur les heuristiques

- heuristiques très rapides et de bonne qualité (théorique et pratique)
- complémentarité des heuristiques
- algorithme génétique ayant un très bon comportement (algorithme hybride basé sur l'heuristique MP)
- point faible :  $1 \ll h_{\min} \simeq h_{\max}$

## Synthèse sur les heuristiques

- heuristiques très rapides et de bonne qualité (théorique et pratique)
- complémentarité des heuristiques
- algorithme génétique ayant un très bon comportement (algorithme hybride basé sur l'heuristique MP)
- point faible :  $1 \ll h_{\min} \simeq h_{\max}$
- besoin de meilleures bornes inférieures

# *Variations*



*Incompatibilités  
entre boîtes et objets*

# Incompatibilités

# Incompatibilités



des documents

# Incompatibilités



des documents

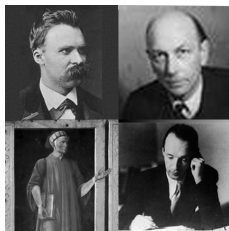


des experts

# Incompatibilités



des documents



des experts

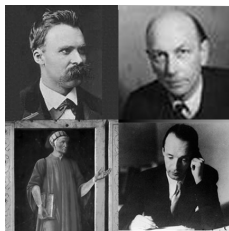
- chaque document doit être analysé par plusieurs experts



# Incompatibilités



des documents



des experts

- chaque document doit être analysé par plusieurs experts
- les experts ne sont compétents que pour certains documents

# Incompatibilités



des documents



des experts

- chaque document doit être analysé par plusieurs experts
- les experts ne sont compétents que pour certains documents
- durées d'expertise unitaires

Modélisation de  $B|allow_j, h_j = 1|H_{\max}$

Modélisation de  $B|allow_j, h_j = 1|H_{\max}$ 

- objet  $O_j$  avec  $s_j$  copies

Modélisation de  $B|allow_j, h_j = 1|H_{\max}$ 

- objet  $O_j$  avec  $s_j$  copies  $\rightarrow$  sommet-objet, poids  $s_j$

2 (a)

(c) 1

2 (b)

(d) 1

Modélisation de  $B|allow_j, h_j = 1|H_{\max}$ 

- objet  $O_j$  avec  $s_j$  copies  $\rightarrow$  sommet-objet, poids  $s_j$
- boîte  $B_i$

2 (a)

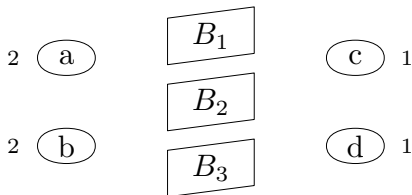
(c) 1

2 (b)

(d) 1

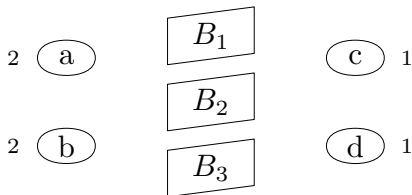
Modélisation de  $B|allow_j, h_j = 1|H_{\max}$ 

- objet  $O_j$  avec  $s_j$  copies  $\rightarrow$  sommet-objet, poids  $s_j$
- boîte  $B_i$   $\rightarrow$  sommet-boîte



Modélisation de  $B|allow_j, h_j = 1|H_{\max}$ 

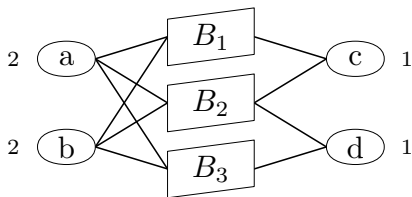
- objet  $O_j$  avec  $s_j$  copies  $\rightarrow$  sommet-objet, poids  $s_j$
- boîte  $B_i$   $\rightarrow$  sommet-boîte
- $O_j$  et  $B_i$  compatibles





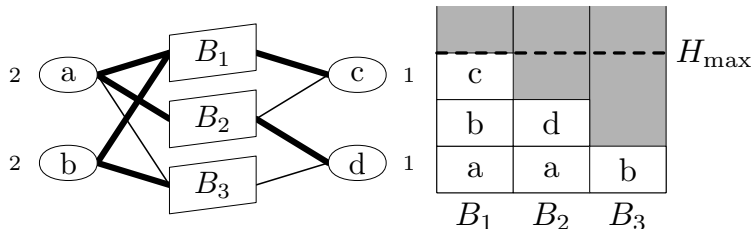
Modélisation de  $B|allow_j, h_j = 1|H_{\max}$ 

- objet  $O_j$  avec  $s_j$  copies  $\rightarrow$  sommet-objet, poids  $s_j$
- boîte  $B_i$   $\rightarrow$  sommet-boîte
- $O_j$  et  $B_i$  compatibles  $\rightarrow$  arête



Modélisation de  $B|allow_j, h_j = 1|H_{\max}$ 

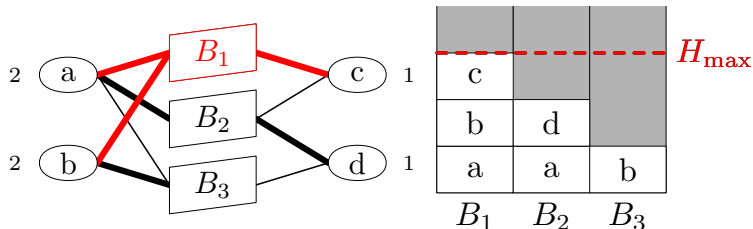
- objet  $O_j$  avec  $s_j$  copies  $\rightarrow$  sommet-objet, poids  $s_j$
- boîte  $B_i$   $\rightarrow$  sommet-boîte
- $O_j$  et  $B_i$  compatibles  $\rightarrow$  arête



- solution : ensemble d'arêtes tel que  $\deg(j) = s_j, \forall j$

Modélisation de  $B|allow_j, h_j = 1|H_{\max}$ 

- objet  $O_j$  avec  $s_j$  copies  $\rightarrow$  sommet-objet, poids  $s_j$
- boîte  $B_i$   $\rightarrow$  sommet-boîte
- $O_j$  et  $B_i$  compatibles  $\rightarrow$  arête

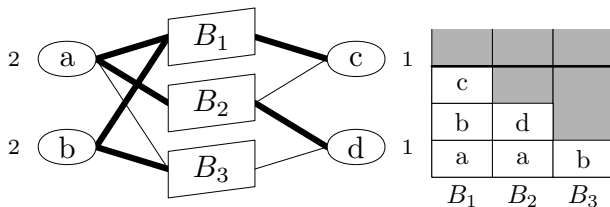


- solution : ensemble d'arêtes tel que  $\deg(j) = s_j, \forall j$
- valeur :  $H_{\max} = \max_i \deg(i)$

## Principe de résolution

## Algorithme d'amélioration

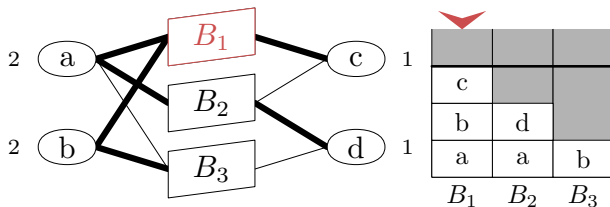
- ❶ choisir une boîte  $B_{i_1}$  « pleine » ( $H_{i_1} = H_{\max}$ )
- ❷ enlever un objet de  $B_{i_1}$
- ❸ placer l'objet dans une boîte  $B_{i_2}$  compatible
- ❹ si  $H_{i_2} < H_{\max}$  alors STOP  
sinon retourner en 2 avec  $i_1 = i_2$



## Principe de résolution

## Algorithme d'amélioration

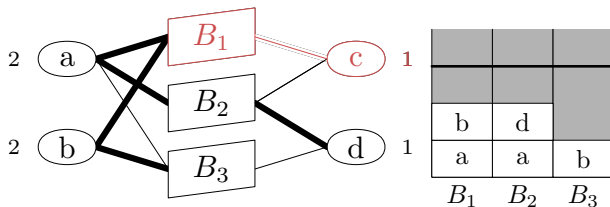
- ❶ choisir une boîte  $B_{i_1}$  « pleine » ( $H_{i_1} = H_{\max}$ )
- ❷ enlever un objet de  $B_{i_1}$
- ❸ placer l'objet dans une boîte  $B_{i_2}$  compatible
- ❹ si  $H_{i_2} < H_{\max}$  alors STOP  
sinon retourner en 2 avec  $i_1 = i_2$



## Principe de résolution

## Algorithme d'amélioration

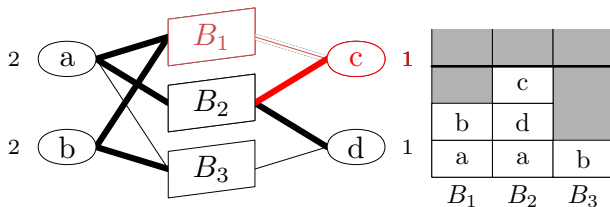
- ❶ choisir une boîte  $B_{i_1}$  « pleine » ( $H_{i_1} = H_{\max}$ )
- ❷ enlever un objet de  $B_{i_1}$
- ❸ placer l'objet dans une boîte  $B_{i_2}$  compatible
- ❹ si  $H_{i_2} < H_{\max}$  alors STOP  
sinon retourner en 2 avec  $i_1 = i_2$



## Principe de résolution

## Algorithme d'amélioration

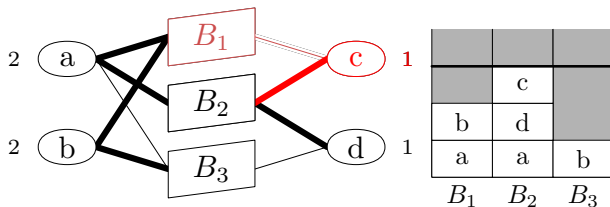
- 1 choisir une boîte  $B_{i_1}$  « pleine » ( $H_{i_1} = H_{\max}$ )
- 2 enlever un objet de  $B_{i_1}$
- 3 placer l'objet dans une boîte  $B_{i_2}$  compatible
- 4 si  $H_{i_2} < H_{\max}$  alors STOP  
sinon retourner en 2 avec  $i_1 = i_2$



## Principe de résolution

## Algorithme d'amélioration

- 1 choisir une boîte  $B_{i_1}$  « pleine » ( $H_{i_1} = H_{\max}$ )
- 2 enlever un objet de  $B_{i_1}$
- 3 placer l'objet dans une boîte  $B_{i_2}$  compatible
- 4 si  $H_{i_2} < H_{\max}$  alors STOP  
sinon retourner en 2 avec  $i_1 = i_2$

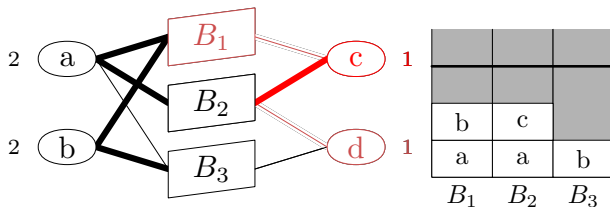




## Principe de résolution

## Algorithme d'amélioration

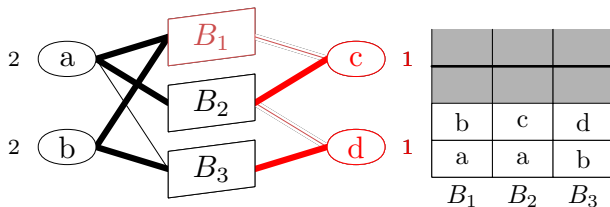
- 1 choisir une boîte  $B_{i_1}$  « pleine » ( $H_{i_1} = H_{\max}$ )
- 2 enlever un objet de  $B_{i_1}$
- 3 placer l'objet dans une boîte  $B_{i_2}$  compatible
- 4 si  $H_{i_2} < H_{\max}$  alors STOP  
sinon retourner en 2 avec  $i_1 = i_2$



## Principe de résolution

## Algorithme d'amélioration

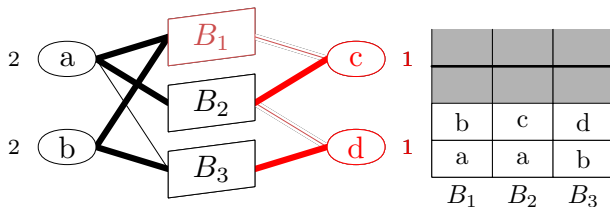
- 1 choisir une boîte  $B_{i_1}$  « pleine » ( $H_{i_1} = H_{\max}$ )
- 2 enlever un objet de  $B_{i_1}$
- 3 placer l'objet dans une boîte  $B_{i_2}$  compatible
- 4 si  $H_{i_2} < H_{\max}$  alors STOP  
sinon retourner en 2 avec  $i_1 = i_2$



## Principe de résolution

## Algorithme d'amélioration

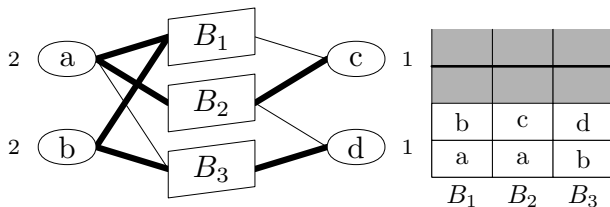
- 1 choisir une boîte  $B_{i_1}$  « pleine » ( $H_{i_1} = H_{\max}$ )
- 2 enlever un objet de  $B_{i_1}$
- 3 placer l'objet dans une boîte  $B_{i_2}$  compatible
- 4 si  $H_{i_2} < H_{\max}$  alors STOP  
sinon retourner en 2 avec  $i_1 = i_2$



## Principe de résolution

## Algorithme d'amélioration

- ❶ choisir une boîte  $B_{i_1}$  « pleine » ( $H_{i_1} = H_{\max}$ )
- ❷ enlever un objet de  $B_{i_1}$
- ❸ placer l'objet dans une boîte  $B_{i_2}$  compatible
- ❹ si  $H_{i_2} < H_{\max}$  alors STOP  
sinon retourner en 2 avec  $i_1 = i_2$



# Calcul d'une solution minimale

## Calcul d'une solution minimale

- construction d'un chemin réducteur en  $\mathcal{O}(|E|)$

## Calcul d'une solution minimale

- construction d'un chemin réducteur en  $\mathcal{O}(|E|)$
- solution minimale :

## Calcul d'une solution minimale

- construction d'un chemin réducteur en  $\mathcal{O}(|E|)$
- solution minimale :
  - de valeur optimale ( $H_{\max} = H_{\max}^*$ )



## Calcul d'une solution minimale

- construction d'un chemin réducteur en  $\mathcal{O}(|E|)$
- solution minimale :
  - de valeur optimale ( $H_{\max} = H_{\max}^*$ )
  - avec le minimum de boîtes de hauteur  $H_{\max}^*$

## Calcul d'une solution minimale

- construction d'un chemin réducteur en  $\mathcal{O}(|E|)$
- solution minimale :
  - de valeur optimale ( $H_{\max} = H_{\max}^*$ )
  - avec le minimum de boîtes de hauteur  $H_{\max}^*$

### théorème

la solution est minimale  $\iff$  il n'existe pas de chemin réducteur

## Calcul d'une solution minimale

- construction d'un chemin réducteur en  $\mathcal{O}(|E|)$
- solution minimale :
  - de valeur optimale ( $H_{\max} = H_{\max}^*$ )
  - avec le minimum de boîtes de hauteur  $H_{\max}^*$

### théorème

la solution est minimale  $\iff$  il n'existe pas de chemin réducteur

- résolution optimale de  $B|allow_j, h_j = 1|H_{\max}$  en  $\mathcal{O}((mn)^2)$

*Final*

A musical staff with five lines, tilted upwards from left to right. It contains ten dark grey oval notes, each with a vertical stem. The notes are arranged in a sequence that rises and then falls. A thick grey bar is positioned below the first five notes, and another thick grey bar is positioned below the last five notes.

*Conclusions et perspectives  
de ce travail*

A musical staff with five lines, tilted upwards from left to right. It contains ten dark grey oval notes, each with a vertical stem. The notes are arranged in a sequence that rises and then falls. A thick grey bar is positioned below the first five notes, and another thick grey bar is positioned below the last five notes.

# Étude générale des problèmes multiboîtes

# Étude générale des problèmes multiboîtes

- présentation du concept d'« objet multiboîte »

# Étude générale des problèmes multiboîtes

- présentation du concept d'« objet multiboîte »
  - formalisation du concept

# Étude générale des problèmes multiboîtes

- présentation du concept d'« objet multiboîte »
  - formalisation du concept
  - classification des modèles



# Étude générale des problèmes multiboîtes

- présentation du concept d'« objet multiboîte »
  - formalisation du concept
  - classification des modèles
- étude de la complexité

# Étude générale des problèmes multiboîtes

- présentation du concept d'« objet multiboîte »
  - formalisation du concept
  - classification des modèles
- étude de la complexité
  - $B|\dots|\cdot$   $\mathcal{NP}$ -complets au sens fort

# Étude générale des problèmes multiboîtes

- présentation du concept d'« objet multiboîte »
  - formalisation du concept
  - classification des modèles
- étude de la complexité
  - $B|\dots|\cdot$   $\mathcal{NP}$ -complets au sens fort
  - $Bm|\dots|\cdot$   $\mathcal{NP}$ -complets au sens ordinaire

# Étude générale des problèmes multiboîtes

- présentation du concept d'« objet multiboîte »
  - formalisation du concept
  - classification des modèles
- étude de la complexité
  - $B|\dots|\cdot \mathcal{NP}$ -complets au sens fort
  - $Bm|\dots|\cdot \mathcal{NP}$ -complets au sens ordinaire
  - $B|size_j, h_j = 1|\cdot$  polynomiaux

# Étude générale des problèmes multiboîtes

- présentation du concept d'« objet multiboîte »
  - formalisation du concept
  - classification des modèles
- étude de la complexité
  - $B|\dots|\cdot \mathcal{NP}$ -complets au sens fort
  - $Bm|\dots|\cdot \mathcal{NP}$ -complets au sens ordinaire
  - $B|size_j, h_j = 1|\cdot$  polynomiaux
  - $B|allow_j, h_j = 1|H_{\max}$  polynomial

# Étude générale des problèmes multiboîtes

- présentation du concept d'« objet multiboîte »
  - formalisation du concept
  - classification des modèles
- étude de la complexité
  - $B|\dots|\cdot \mathcal{NP}$ -complets au sens fort
  - $Bm|\dots|\cdot \mathcal{NP}$ -complets au sens ordinaire
  - $B|size_j, h_j = 1|\cdot$  polynomiaux
  - $B|allow_j, h_j = 1|H_{\max}$  polynomial
- étude de l'approximabilité

# Étude générale des problèmes multiboîtes

- présentation du concept d'« objet multiboîte »
  - formalisation du concept
  - classification des modèles
- étude de la complexité
  - $B|\dots|\cdot \mathcal{NP}$ -complets au sens fort
  - $Bm|\dots|\cdot \mathcal{NP}$ -complets au sens ordinaire
  - $B|size_j, h_j = 1|\cdot$  polynomiaux
  - $B|allow_j, h_j = 1|H_{\max}$  polynomial
- étude de l'approximabilité
  - existence de FPTAS pour  $Bm|any_j|H_{\max}$

# Étude de $Bm|size_j|H_{\max}$



## Étude de $Bm|size_j|H_{\max}$

- proposition de plusieurs heuristiques

## Étude de $Bm|size_j|H_{\max}$

- proposition de plusieurs heuristiques
  - avec des garanties théoriques

## Étude de $Bm|size_j|H_{\max}$

- proposition de plusieurs heuristiques
  - avec des garanties théoriques
  - rapides et efficaces en pratique

## Étude de $Bm|size_j|H_{\max}$

- proposition de plusieurs heuristiques
  - avec des garanties théoriques
  - rapides et efficaces en pratique
  - complémentaires

## Étude de $Bm|size_j|H_{\max}$

- proposition de plusieurs heuristiques
  - avec des garanties théoriques
  - rapides et efficaces en pratique
  - complémentaires
- proposition d'un algorithme génétique

## Étude de $Bm|size_j|H_{\max}$

- proposition de plusieurs heuristiques
  - avec des garanties théoriques
  - rapides et efficaces en pratique
  - complémentaires
- proposition d'un algorithme génétique
  
- résolution d'instances aléatoires

## Étude de $Bm|size_j|H_{\max}$

- proposition de plusieurs heuristiques
  - avec des garanties théoriques
  - rapides et efficaces en pratique
  - complémentaires
- proposition d'un algorithme génétique
  
- résolution d'instances aléatoires
  - de très bonnes solutions rapidement

## Étude de $Bm|size_j|H_{\max}$

- proposition de plusieurs heuristiques
  - avec des garanties théoriques
  - rapides et efficaces en pratique
  - complémentaires
- proposition d'un algorithme génétique
  
- résolution d'instances aléatoires
  - de très bonnes solutions rapidement
  - la plupart des instances sont résolues optimalement



## Étude de $Bm|size_j|H_{\max}$

- proposition de plusieurs heuristiques
  - avec des garanties théoriques
  - rapides et efficaces en pratique
  - complémentaires
- proposition d'un algorithme génétique
  
- résolution d'instances aléatoires
  - de très bonnes solutions rapidement
  - la plupart des instances sont résolues optimalement
  - peu d'instances mal résolues

## Quelques pistes de recherche

## Quelques pistes de recherche

- étudier en détails les différents modèles

## Quelques pistes de recherche

- étudier en détails les différents modèles
  - objectif  $m$  (*cf bin-packing*)

## Quelques pistes de recherche

- étudier en détails les différents modèles
  - objectif  $m$  (*cf bin-packing*)
  - objectif  $N$  (*cf bin-packing* de cardinalité maximum)

## Quelques pistes de recherche

- étudier en détails les différents modèles
  - objectif  $m$  (cf *bin-packing*)
  - objectif  $N$  (cf *bin-packing* de cardinalité maximum)
  - cas de boîtes non identiques

## Quelques pistes de recherche

- étudier en détails les différents modèles
  - objectif  $m$  (cf *bin-packing*)
  - objectif  $N$  (cf *bin-packing* de cardinalité maximum)
  - cas de boîtes non identiques
- méthodes exactes

## Quelques pistes de recherche

- étudier en détails les différents modèles
  - objectif  $m$  (cf *bin-packing*)
  - objectif  $N$  (cf *bin-packing* de cardinalité maximum)
  - cas de boîtes non identiques
- méthodes exactes
  - recherches de méthodes exactes efficaces



## Quelques pistes de recherche

- étudier en détails les différents modèles
  - objectif  $m$  (cf *bin-packing*)
  - objectif  $N$  (cf *bin-packing* de cardinalité maximum)
  - cas de boîtes non identiques
- méthodes exactes
  - recherches de méthodes exactes efficaces
  - études de bonnes bornes inférieures

## Quelques pistes de recherche

- étudier en détails les différents modèles
  - objectif  $m$  (cf *bin-packing*)
  - objectif  $N$  (cf *bin-packing* de cardinalité maximum)
  - cas de boîtes non identiques
- méthodes exactes
  - recherches de méthodes exactes efficaces
  - études de bonnes bornes inférieures
- approximations

## Quelques pistes de recherche

- étudier en détails les différents modèles
  - objectif  $m$  (*cf bin-packing*)
  - objectif  $N$  (*cf bin-packing* de cardinalité maximum)
  - cas de boîtes non identiques
- méthodes exactes
  - recherches de méthodes exactes efficaces
  - études de bonnes bornes inférieures
- approximations
  - études de variations de DR

## Quelques pistes de recherche

- étudier en détails les différents modèles
  - objectif  $m$  (*cf bin-packing*)
  - objectif  $N$  (*cf bin-packing* de cardinalité maximum)
  - cas de boîtes non identiques
- méthodes exactes
  - recherches de méthodes exactes efficaces
  - études de bonnes bornes inférieures
- approximations
  - études de variations de DR
- étude pratique

## Quelques pistes de recherche

- étudier en détails les différents modèles
  - objectif  $m$  (*cf bin-packing*)
  - objectif  $N$  (*cf bin-packing* de cardinalité maximum)
  - cas de boîtes non identiques
- méthodes exactes
  - recherches de méthodes exactes efficaces
  - études de bonnes bornes inférieures
- approximations
  - études de variations de DR
- étude pratique
  - améliorations de l'algorithme génétique

## Quelques pistes de recherche

- étudier en détails les différents modèles
  - objectif  $m$  (cf *bin-packing*)
  - objectif  $N$  (cf *bin-packing* de cardinalité maximum)
  - cas de boîtes non identiques
- méthodes exactes
  - recherches de méthodes exactes efficaces
  - études de bonnes bornes inférieures
- approximations
  - études de variations de DR
- étude pratique
  - améliorations de l'algorithme génétique
  - résolution de problèmes réels

## Question

existe-t-il toujours un ordre des objets tel que l'algorithme de meilleur placement produit un rangement optimal pour  $B|size_j|H_{\max}$  ?







# Les algorithmes génétiques

# Les algorithmes génétiques

- population : ensemble de chromosomes (solutions)

# Les algorithmes génétiques

- population : ensemble de chromosomes (solutions)

## Algorithme génétique

- 1 sélection
- 2 reproduction
- 3 évaluation

# Les algorithmes génétiques

- population : ensemble de chromosomes (solutions)

## Algorithme génétique

- 1 sélection
- 2 reproduction
- 3 évaluation

- seuls les meilleurs individus sont gardés

# Les algorithmes génétiques

- population : ensemble de chromosomes (solutions)

## Algorithme génétique

- 1 sélection
- 2 reproduction
- 3 évaluation

- les parents sont choisis aléatoirement, selon leur qualité

# Les algorithmes génétiques

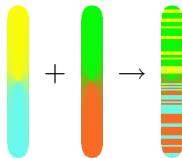
- population : ensemble de chromosomes (solutions)

## Algorithme génétique

- 1 sélection
- 2 reproduction
- 3 évaluation

- les parents sont choisis aléatoirement, selon leur qualité

- croisement



# Les algorithmes génétiques

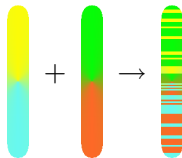
- population : ensemble de chromosomes (solutions)

## Algorithme génétique

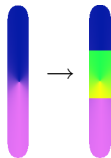
- 1 sélection
- 2 reproduction
- 3 évaluation

- les parents sont choisis aléatoirement, selon leur qualité

- croisement



- mutation





# Les algorithmes génétiques

- population : ensemble de chromosomes (solutions)

## Algorithme génétique

- 1 sélection
- 2 reproduction
- 3 **évaluation**

- les rejetons sont évalués et ajoutés à la population

Un algorithme génétique pour  $B|size_j|H_{\max}$

## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets

## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets
- **évaluation** : valeur de MP pour cet ordre

## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets
- **évaluation** : valeur de MP pour cet ordre

avantages :

- codage simple et efficace
- évaluation rapide

## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets
- **évaluation** : valeur de MP pour cet ordre

### avantages :

- codage simple et efficace
- évaluation rapide

### problème :

- pas de preuve que l'optimum est toujours atteignable (conjecture)

## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets
- **évaluation** : valeur de MP pour cet ordre

### avantages :

- codage simple et efficace
- évaluation rapide

### problème :

- pas de preuve que l'optimum est toujours atteignable (conjecture)
- **mutation** : « *scramble list mutation* »



Davis 91

1 2 3 4 5 6 7 8 9

## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets
- **évaluation** : valeur de MP pour cet ordre

### avantages :

- codage simple et efficace
- évaluation rapide

### problème :

- pas de preuve que l'optimum est toujours atteignable (conjecture)
- **mutation** : « *scramble list mutation* »



Davis 91

1 2 3 4 5 6 7 8 9



## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets
- **évaluation** : valeur de MP pour cet ordre

### avantages :

- codage simple et efficace
- évaluation rapide

### problème :

- pas de preuve que l'optimum est toujours atteignable (conjecture)
- **mutation** : « *scramble list mutation* »



Davis 91

1 2 4 6 5 3 7 8 9

## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets
- **évaluation** : valeur de MP pour cet ordre

### avantages :

- codage simple et efficace
- évaluation rapide

### problème :

- pas de preuve que l'optimum est toujours atteignable (conjecture)
- **mutation** : « *scramble list mutation* »



Davis 91

1 2 4 6 5 3 7 8 9

## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets
- **évaluation** : valeur de MP pour cet ordre

### avantages :

- codage simple et efficace
- évaluation rapide

### problème :

- pas de preuve que l'optimum est toujours atteignable (conjecture)
- **mutation** : « *scramble list mutation* »
- **croisement** : « *uniform orde-based crossover* »



Davis 91

1 2 3 4 5 6 7 8 -- 8 6 4 2 7 5 3 1

## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets
- **évaluation** : valeur de MP pour cet ordre

### avantages :

- codage simple et efficace
- évaluation rapide

### problème :

- pas de preuve que l'optimum est toujours atteignable (conjecture)
- **mutation** : « *scramble list mutation* »
- **croisement** : « *uniform orde-based crossover* »



Davis 91

1 2 3 4 5 6 7 8 -- 8 6 4 2 7 5 3 1

## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets
- **évaluation** : valeur de MP pour cet ordre

### avantages :

- codage simple et efficace
- évaluation rapide

### problème :

- pas de preuve que l'optimum est toujours atteignable (conjecture)
- **mutation** : « *scramble list mutation* »
- **croisement** : « *uniform orde-based crossover* »



Davis 91

. 2 3 . 5 6 . . -- 8,4,7,1

## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets
- **évaluation** : valeur de MP pour cet ordre

### avantages :

- codage simple et efficace
- évaluation rapide

### problème :

- pas de preuve que l'optimum est toujours atteignable (conjecture)
- **mutation** : « *scramble list mutation* »
- **croisement** : « *uniform orde-based crossover* »



Davis 91

8 2 3 4 5 6 7 1



## Un algorithme génétique pour $B|size_j|H_{\max}$

- **chromosome** : permutation de l'ensemble des objets
- **évaluation** : valeur de MP pour cet ordre

### avantages :

- codage simple et efficace
- évaluation rapide

### problème :

- pas de preuve que l'optimum est toujours atteignable (conjecture)
- **mutation** : « *scramble list mutation* »  
 Davis 91
- **croisement** : « *uniform orde-based crossover* »  
 Davis 91

## Apport de l'algorithme génétique

