

# Konstruktive algoritmer og nedre grænseværdier for guillotinesnitbare ortogonale bin packing problemer

## Specialeforsvar

Rasmus Resen Amossen

Datalogisk Institut  
Københavns Universitet

Fredag d. 18/11 2005

# Disposition

## 1 Problemstillinger

# Disposition

1 Problemstillinger

2 Fekete og Schepers' framework

# Disposition

- 1 Problemstillinger
- 2 Fekete og Schepers' framework
- 3 Guillotinesnit

# Disposition

- 1 Problemstillinger
- 2 Fekete og Schepers' framework
- 3 Guillotinesnit
- 4 Stickycutting

# Disposition

- 1 Problemstillinger
- 2 Fekete og Schepers' framework
- 3 Guillotinesnit
- 4 Sticky cutting
- 5 Resultater

# Disposition

- 1 Problemstillinger
- 2 Fekete og Schepers' framework
- 3 Guillotinesnit
- 4 Sticky cutting
- 5 Resultater
- 6 Afrunding

# Disposition

1 Problemstillinger

2 Fekete og Schepers' framework

3 Guillotinesnit

4 Sticky cutting

5 Resultater

6 Afrunding

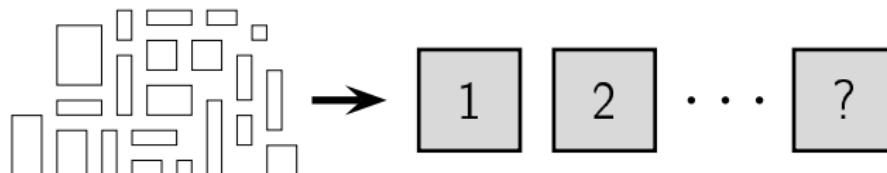
# Problemstillinger

Mængde,  $B$ , af kasser af størrelse  $w : B \rightarrow \mathbb{R}_+^d$ .  
Containerstørrelse  $W \in \mathbb{R}_+^d$ .

# Problemstillinger

Mængde,  $B$ , af kasser af størrelse  $w : B \rightarrow \mathbb{R}_+^d$ .  
Containerstørrelse  $W \in \mathbb{R}_+^d$ .

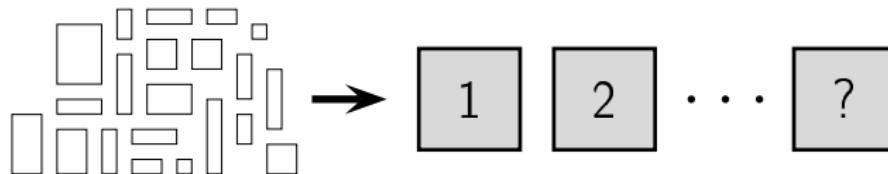
- Orthogonal Bin Packing Problem (OBPP- $d$ )



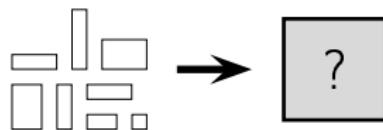
# Problemstillinger

Mængde,  $B$ , af kasser af størrelse  $w : B \rightarrow \mathbb{R}_+^d$ .  
Containerstørrelse  $W \in \mathbb{R}_+^d$ .

- Orthogonal Bin Packing Problem (OBPP- $d$ )



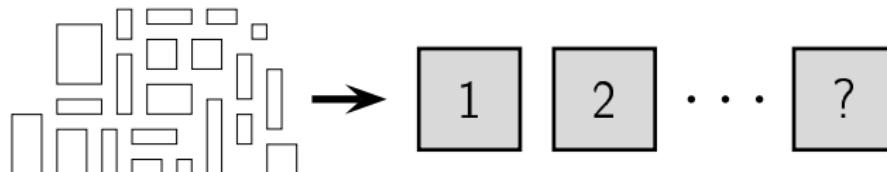
- Orthogonal Packing Problem (OPP- $d$ ).



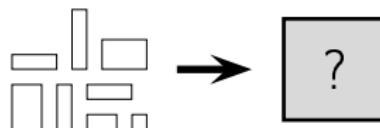
# Problemstillinger

Mængde,  $B$ , af kasser af størrelse  $w : B \rightarrow \mathbb{R}_+^d$ .  
Containerstørrelse  $W \in \mathbb{R}_+^d$ .

- Orthogonal Bin Packing Problem (OBPP- $d$ )



- Orthogonal Packing Problem (OPP- $d$ ).



- Begge er NP-hårde

# Metoder for OBPP- $d$

Branch-and-bound

- Ydre branch-and-bound algoritme distribuerer kasser til containere.
- Indre OPP- $d$  algoritme.
- Afskærer søgerum med polynomielle lower bounds.

# Metoder for OBPP- $d$

## LP-model

# Metoder for OBPP- $d$

## LP-model

$$\begin{aligned}
 \min \quad & c \\
 \text{s.t.} \quad & l_{ij} + l_{ji} + b_{ij} + b_{ji} + p_{ij} + p_{ji} \geq 1 \quad i, j \in B, i < j \\
 & x_i - x_j + Wl_{ij} \leq W - w_i \quad i, j \in B \\
 & y_i - y_j + Hb_{ij} \leq H - b_i \quad i, j \in B \\
 & m_i - m_j + np_{ij} \leq n - 1 \quad i, j \in B \\
 & 0 \leq x_i \leq W - w_i \quad i \in B \\
 & 0 \leq y_i \leq H - h_i \quad i \in B \\
 & 1 \leq m_i \leq c \quad i \in B \\
 & l_{ij}, b_{ij}, p_{ij} \in \{0, 1\} \quad i, j \in B \\
 & x_i, y_i \in \mathbb{R}_0^+ \quad i \in B \\
 & m_i, c \in \mathbb{N}_0 \quad i \in B
 \end{aligned}$$

# Metoder for OBPP- $d$

## LP-model

- Stor symmetri,  $O(|B|^2)$  constraints,  $O(|B|^2)$  binære variable.

# Metoder for OBPP- $d$

## LP-model

- Stor symmetri,  $O(|B|^2)$  constraints,  $O(|B|^2)$  binære variable.
- Pisinger og Sigurd:
  - Bruger Dantzig-Wolfe dekomposition og kolonnegenerering.  
Restricted Master Problem: Setcover

$$\begin{aligned} \min \quad & \sum_{p \in \mathcal{P}} x_p \\ \text{s.t.} \quad & \sum_{p \in \mathcal{P}} x_p \delta_i^p \geq 1 \quad i \in B \\ & x_p \in \{0, 1\} \quad p \in \mathcal{P} \end{aligned}$$

Pricing problem:  $d$ -dimensional knapsack

# Metoder for OBPP- $d$

LP-model

- Stor symmetri,  $O(|B|^2)$  constraints,  $O(|B|^2)$  binære variable.
- Pisinger og Sigurd:
  - Bruger Dantzig-Wolfe dekomposition og kolonnegenerering.  
Restricted Master Problem: Setcover

$$\begin{aligned} \min \quad & \sum_{p \in \mathcal{P}} x_p \\ \text{s.t.} \quad & \sum_{p \in \mathcal{P}} x_p \delta_i^p \geq 1 \quad i \in B \\ & x_p \in \{0, 1\} \quad p \in \mathcal{P} \end{aligned}$$

Pricing problem:  $d$ -dimensional knapsack

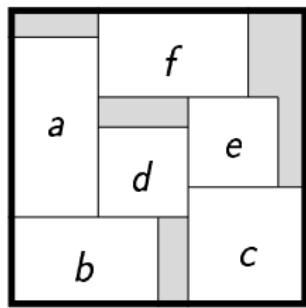
- $d$ -dimensional knapsack  $\longrightarrow$   
1-dimensional knapsack og  $d$ -dimensional OPP- $d$

# Disposition

- 1 Problemstillinger
- 2 Fekete og Schepers' framework
- 3 Guillotinesnit
- 4 Sticky cutting
- 5 Resultater
- 6 Afrunding

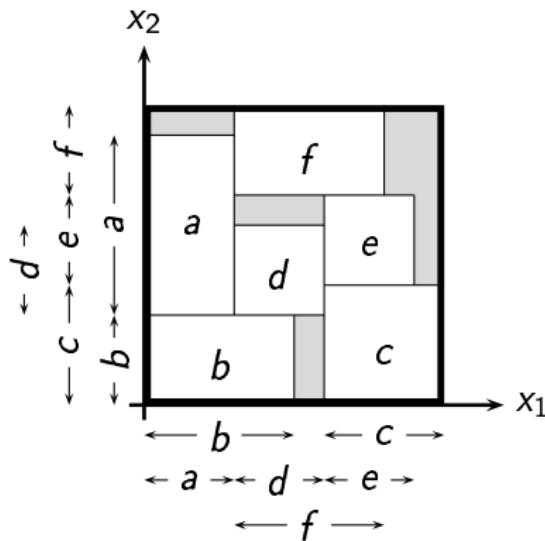
# Modellering

Overordnet idé



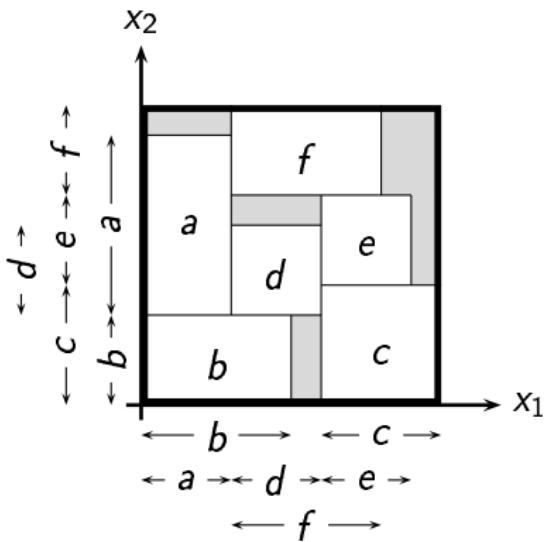
# Modellering

Overordnet idé

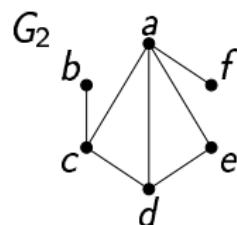
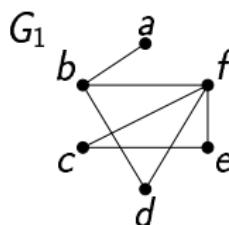


# Modellering

Overordnet idé

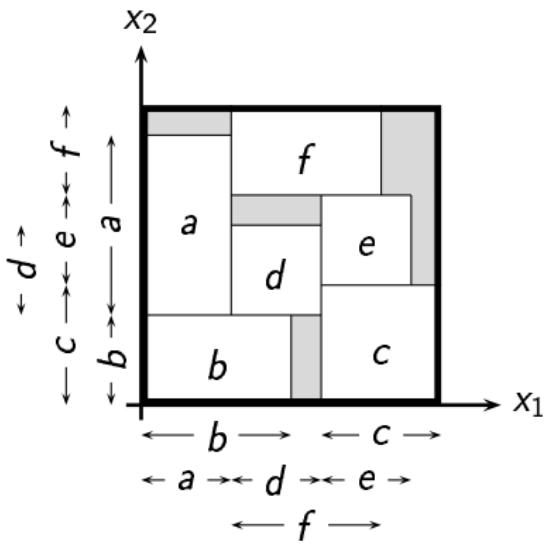


Intervalgrafer:

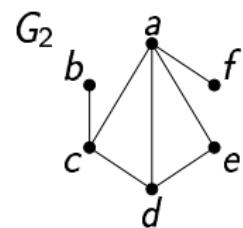
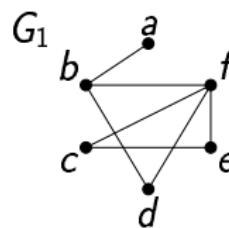


# Modellering

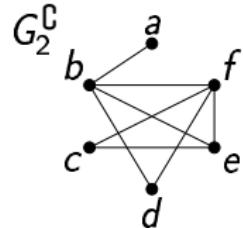
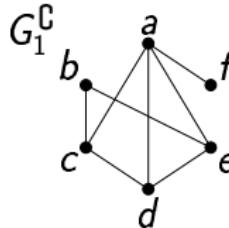
Overordnet idé



Intervalgrafer:

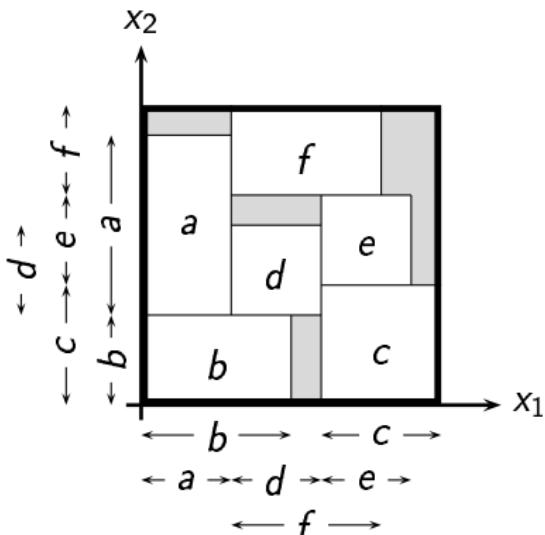


Komplementære grafer:

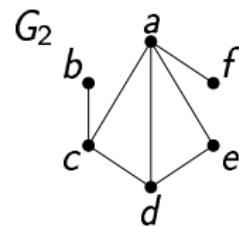
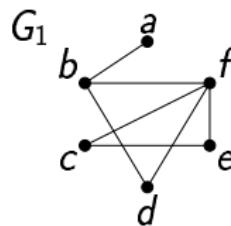


# Modellering

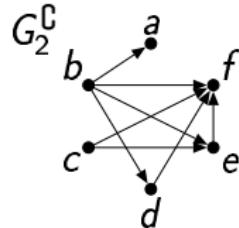
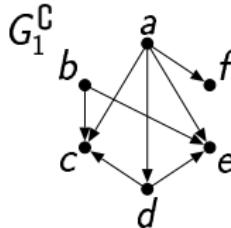
Overordnet idé



Intervalgrafer:

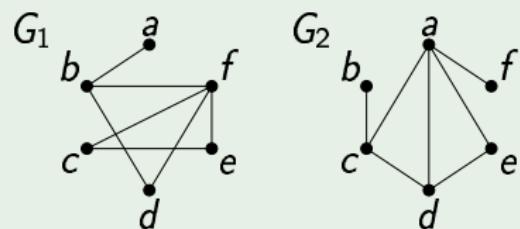
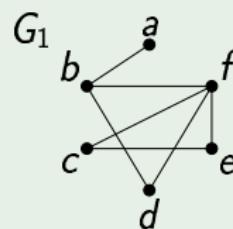
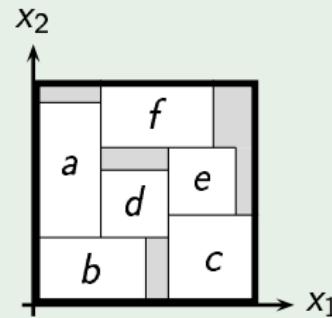


Komplementære grafer:



## P1, P2 og P3

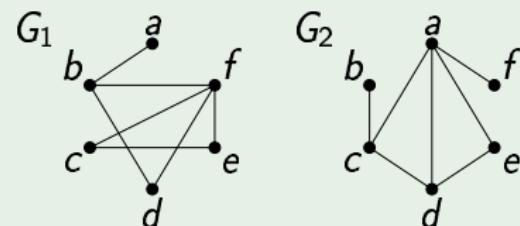
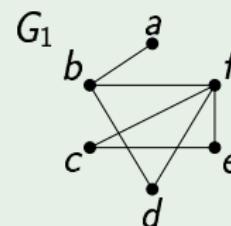
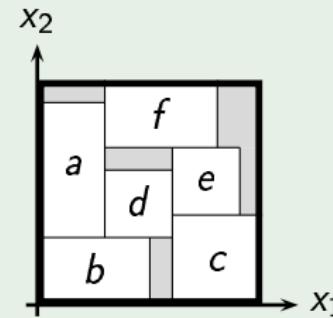
## Eksempel



## P1, P2 og P3

P1  $\forall i \in \{1, \dots, d\}$ :  
 $G_i = (V, E_i)$  intervalgraf

## Eksempel



## P1, P2 og P3

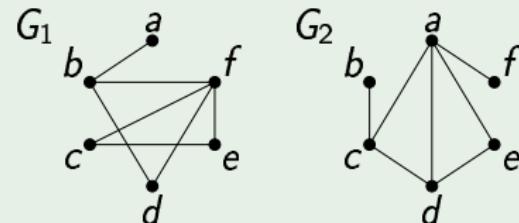
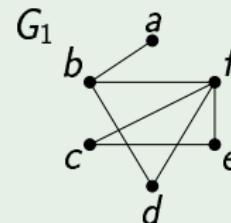
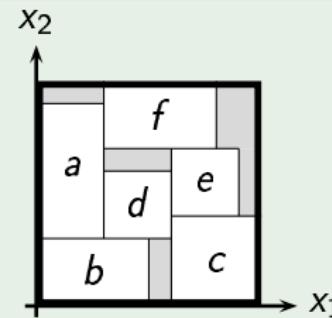
P1  $\forall i \in \{1, \dots, d\}$ :  
 $G_i = (V, E_i)$  intervalgraf

## Definition

$V' \subseteq V$  er  $x_i$ -feasible hvis

$$\sum_{v \in V'} w_i(v) \leq W_i$$

## Eksempel



# P1, P2 og P3

**P1**  $\forall i \in \{1, \dots, d\}$ :  
 $G_i = (V, E_i)$  intervalgraf

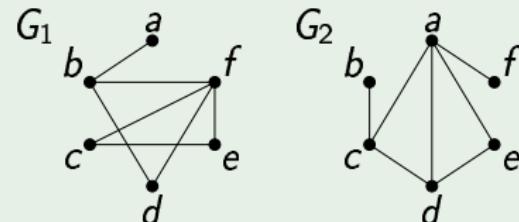
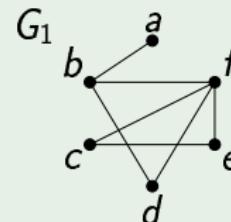
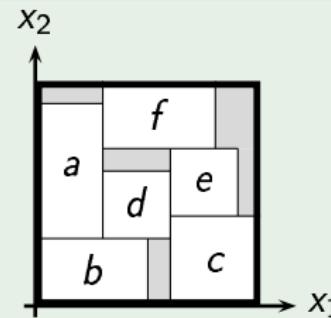
## Definition

$V' \subseteq V$  er  $x_i$ -feasible hvis

$$\sum_{v \in V'} w_i(v) \leq W_i$$

**P2**  $\forall i \in \{1, \dots, d\}$ : Ethvert stable set i  $G_i$  er  $x_i$ -feasible

## Eksempel



# P1, P2 og P3

**P1**  $\forall i \in \{1, \dots, d\}$ :  
 $G_i = (V, E_i)$  intervalgraf

## Definition

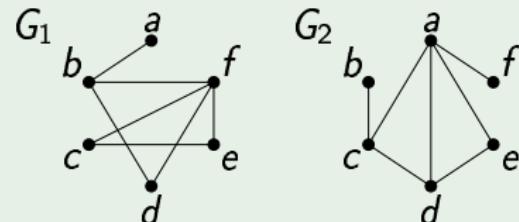
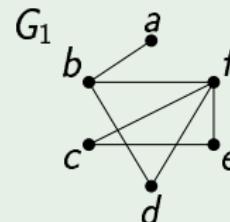
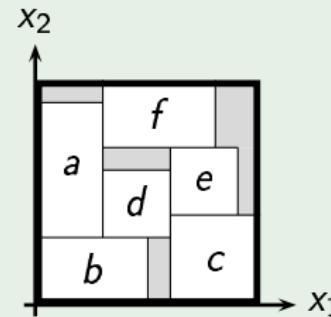
$V' \subseteq V$  er  $x_i$ -feasible hvis

$$\sum_{v \in V'} w_i(v) \leq W_i$$

**P2**  $\forall i \in \{1, \dots, d\}$ : Ethvert stable set i  $G_i$  er  $x_i$ -feasible

**P3**  $\bigcap_{i=1}^d E_i = \emptyset$

## Eksempel



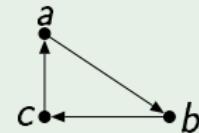
# P1, P2 og P3

## Sætning

$G_1, \dots, G_d$  udgør en pakningsklasse hvis og kun hvis P1, P2 og P3 er opfyldt.

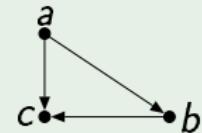
# Terminologi

## Eksempel



# Terminologi

## Eksempel

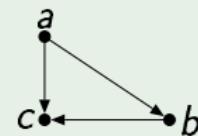


# Terminologi

## Definition

$G$  er en **comparabilitets graf** hvis og kun hvis der findes en transitiv orientering af  $G$ .

## Eksempel

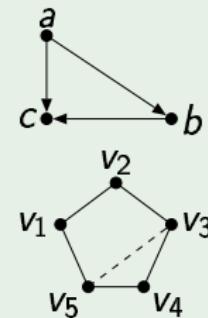


# Terminologi

## Definition

$G$  er en **comparabilitets graf** hvis og kun hvis der findes en transitiv orientering af  $G$ .

## Eksempel

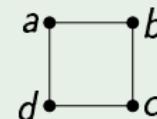
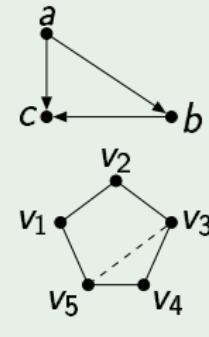


# Terminologi

## Definition

$G$  er en **comparabilitets graf** hvis og kun hvis der findes en transitiv orientering af  $G$ .

## Eksempel



# Terminologi

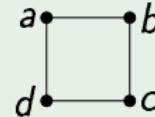
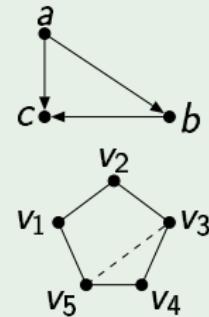
## Definition

*G er en comparabilitets graf hvis og kun hvis der findes en transitiv orientering af G.*

## Sætning

*G er en intervalgraf hvis og kun hvis G ikke indeholder en kordeløs  $C_4$  og  $G^C$  er en comparabilitet graf.*

## Eksempel



# Terminologi

## Definition

$G$  er en **comparabilitets graf** hvis og kun hvis der findes en transitiv orientering af  $G$ .

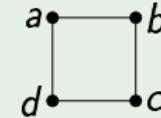
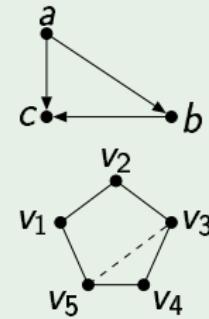
## Sætning

$G$  er en **intervalgraf** hvis og kun hvis  $G$  ikke indeholder en kordeløs  $C_4$  og  $G^C$  er en comparabilitet graf.

## Sætning

$G$  er en **comparabilites graf** hvis og kun hvis den ikke indeholder en 2-kordeløs ulige kreds.

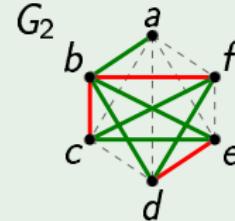
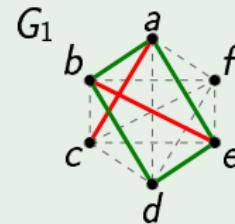
## Eksempel



# Branch-and-bound algoritme

- Fix kanter i  $\mathcal{E}_- = (\mathcal{E}_{-,1}, \dots, \mathcal{E}_{-,d})$  og  $\mathcal{E}_+ = (\mathcal{E}_{+,1}, \dots, \mathcal{E}_{+,d})$

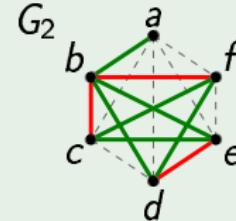
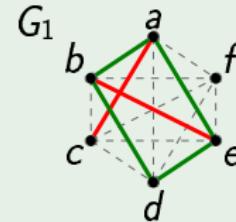
## Eksempel



# Branch-and-bound algoritme

- Fix kanter i  $\mathcal{E}_- = (\mathcal{E}_{-,1}, \dots, \mathcal{E}_{-,d})$  og  $\mathcal{E}_+ = (\mathcal{E}_{+,1}, \dots, \mathcal{E}_{+,d})$
- Branch på  $(e, i, +/-)$

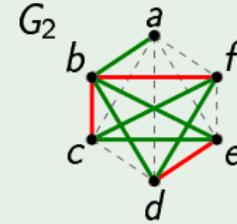
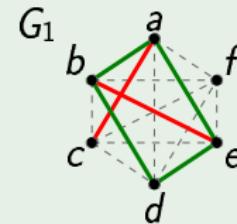
## Eksempel



# Branch-and-bound algoritme

- Fix kanter i  $\mathcal{E}_- = (\mathcal{E}_{-,1}, \dots, \mathcal{E}_{-,d})$  og  $\mathcal{E}_+ = (\mathcal{E}_{+,1}, \dots, \mathcal{E}_{+,d})$
- Branch på  $(e, i, +/-)$
- Tjek løbende for

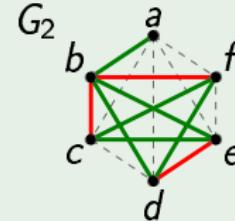
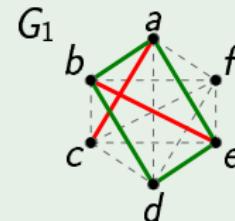
## Eksempel



# Branch-and-bound algoritme

- Fix kanter i  $\mathcal{E}_- = (\mathcal{E}_{-,1}, \dots, \mathcal{E}_{-,d})$  og  $\mathcal{E}_+ = (\mathcal{E}_{+,1}, \dots, \mathcal{E}_{+,d})$
- Branch på  $(e, i, +/-)$
- Tjek løbende for  
P1 Kordeløs  $C_4$  i  $G_i = (V, \mathcal{E}_{+,i})$

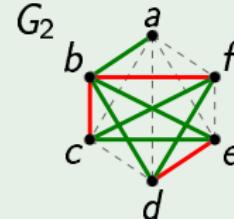
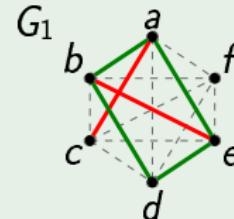
## Eksempel



# Branch-and-bound algoritme

- Fix kanter i  $\mathcal{E}_- = (\mathcal{E}_{-,1}, \dots, \mathcal{E}_{-,d})$  og  $\mathcal{E}_+ = (\mathcal{E}_{+,1}, \dots, \mathcal{E}_{+,d})$
- Branch på  $(e, i, +/-)$
- Tjek løbende for
  - P1 Kordeløs  $C_4$  i  $G_i = (V, \mathcal{E}_{+,i})$
  - P1 2-kordeløs ulige kreds i  $G_i^C = (V, \mathcal{E}_{+,i}^C)$

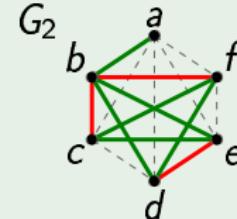
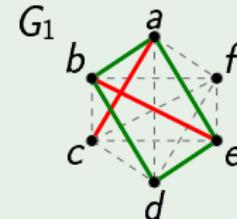
## Eksempel



# Branch-and-bound algoritme

- Fix kanter i  $\mathcal{E}_- = (\mathcal{E}_{-,1}, \dots, \mathcal{E}_{-,d})$  og  $\mathcal{E}_+ = (\mathcal{E}_{+,1}, \dots, \mathcal{E}_{+,d})$
- Branch på  $(e, i, +/-)$
- Tjek løbende for
  - P1 Kordeløs  $C_4$  i  $G_i = (V, \mathcal{E}_{+,i})$
  - P1 2-kordeløs ulige kreds i  $G_i^C = (V, \mathcal{E}_{+,i}^C)$
  - P2  $x_i$ -infeasible klike i  $G_i^C = (V, \mathcal{E}_{+,i}^C)$

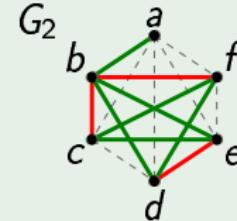
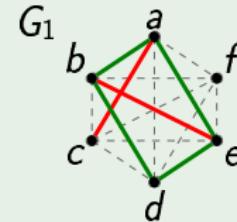
## Eksempel



# Branch-and-bound algoritme

- Fix kanter i  $\mathcal{E}_- = (\mathcal{E}_{-,1}, \dots, \mathcal{E}_{-,d})$  og  $\mathcal{E}_+ = (\mathcal{E}_{+,1}, \dots, \mathcal{E}_{+,d})$
- Branch på  $(e, i, +/-)$
- Tjek løbende for
  - P1 Kordeløs  $C_4$  i  $G_i = (V, \mathcal{E}_{+,i})$
  - P1 2-kordeløs ulige kreds i  $G_i^C = (V, \mathcal{E}_{+,i}^C)$
  - P2  $x_i$ -infeasible klike i  $G_i^C = (V, \mathcal{E}_{+,i}^C)$
  - P3  $\bigcap_{i=1}^d \mathcal{E}_{+,i} = \emptyset$

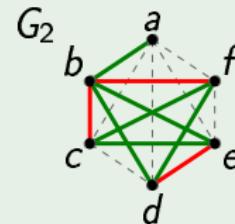
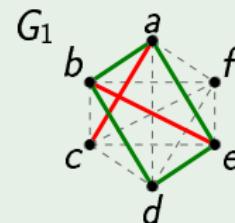
## Eksempel



# Branch-and-bound algoritme

- Fix kanter i  $\mathcal{E}_- = (\mathcal{E}_{-,1}, \dots, \mathcal{E}_{-,d})$  og  $\mathcal{E}_+ = (\mathcal{E}_{+,1}, \dots, \mathcal{E}_{+,d})$
- Branch på  $(e, i, +/-)$
- Tjek løbende for
  - P1 Kordeløs  $C_4$  i  $G_i = (V, \mathcal{E}_{+,i})$
  - P1 2-kordeløs ulige kreds i  $G_i^C = (V, \mathcal{E}_{+,i}^C)$
  - P2  $x_i$ -infeasible klike i  $G_i^C = (V, \mathcal{E}_{+,i}^C)$
  - P3  $\bigcap_{i=1}^d \mathcal{E}_{+,i} = \emptyset$
- Ingen brud:  
 $G_1 = (V, \mathcal{E}_{+,1}), \dots, G_d = (V, \mathcal{E}_{+,d})$  udgør pakningsklasse

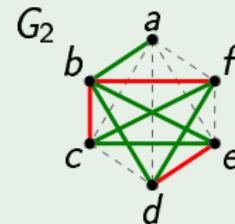
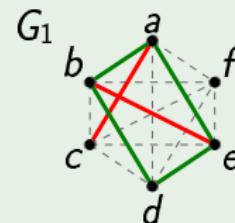
## Eksempel



# Branch-and-bound algoritme

- Fix kanter i  $\mathcal{E}_- = (\mathcal{E}_{-,1}, \dots, \mathcal{E}_{-,d})$  og  $\mathcal{E}_+ = (\mathcal{E}_{+,1}, \dots, \mathcal{E}_{+,d})$
- Branch på  $(e, i, +/-)$
- Tjek løbende for
  - P1 Kordeløs  $C_4$  i  $G_i = (V, \mathcal{E}_{+,i})$
  - P1 2-kordeløs ulige kreds i  $G_i^C = (V, \mathcal{E}_{+,i}^C)$
  - P2  $x_i$ -infeasible klike i  $G_i^C = (V, \mathcal{E}_{+,i}^C)$
  - P3  $\bigcap_{i=1}^d \mathcal{E}_{+,i} = \emptyset$
- Ingen brud:  
 $G_1 = (V, \mathcal{E}_{+,1}), \dots, G_d = (V, \mathcal{E}_{+,d})$  udgør pakningsklasse
- Pakningsisomorfier ikke beskrevet

## Eksempel



# Disposition

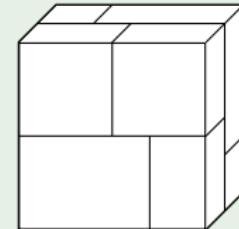
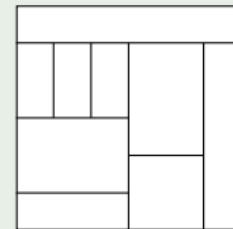
- 1 Problemstillinger
- 2 Fekete og Schepers' framework
- 3 Guillotinesnit
- 4 Sticky cutting
- 5 Resultater
- 6 Afrunding

# Terminologi

## Definition

$d - 1$  dimensionelt akseparallelt hyperplan,  $\mathcal{P}$ ,  
**guillotinesnit** hvis  $\mathcal{P}$  deler  $U \subseteq V$  i to  
disjunkte ikke-tomme mængder så ingen  $u \in U$   
skæres af  $\mathcal{P}$ .

## Eksempel



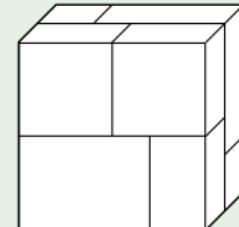
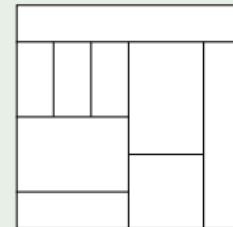
# Terminologi

## Definition

$d - 1$  dimensionelt akseparallelt hyperplan,  $\mathcal{P}$ ,  
**guillotinesnit** hvis  $\mathcal{P}$  deler  $U \subseteq V$  i to  
disjunkte ikke-tomme mængder så ingen  $u \in U$   
skæres af  $\mathcal{P}$ .

Snit foregår i **trin/stages**, hvor alle snit af  
cutslice  $U$  i samme trin er parallelle.

## Eksempel



# Terminologi

## Definition

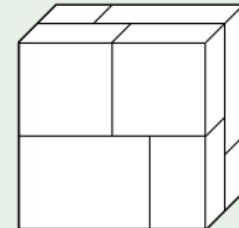
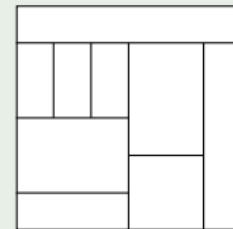
$d - 1$  dimensionelt akseparallelt hyperplan,  $\mathcal{P}$ , **guillotinesnit** hvis  $\mathcal{P}$  deler  $U \subseteq V$  i to disjunkte ikke-tomme mængder så ingen  $u \in U$  skæres af  $\mathcal{P}$ .

Snit foregår i **trin/stages**, hvor alle snit af cutslice  $U$  i samme trin er parallelle.

## Definition

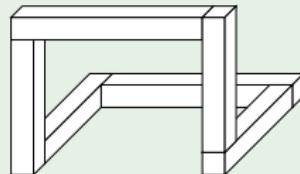
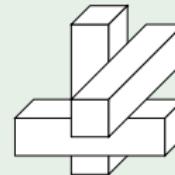
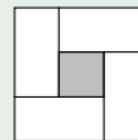
$p$  er en **k-stage pakning** hvis alle cutslices er singleton i k'te rekursionstrin.

## Eksempel



# Egenskaber

## Eksempel (Blokeret ring)

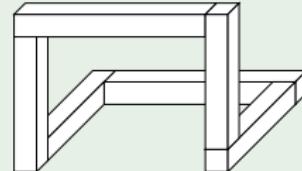
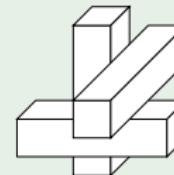
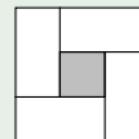


# Egenskaber

## Sætning

*p guillotinesnitbar hvis og kun hvis der ikke findes en blokeret ring i p.*

## Eksempel (Blokeret ring)



# Egenskaber

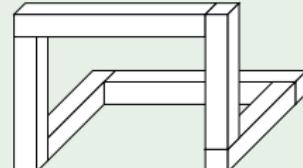
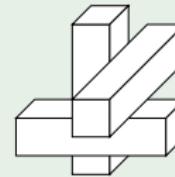
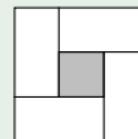
## Sætning

*p guillotinesnitbar hvis og kun hvis der ikke findes en blokeret ring i p.*

## Sætning

*For  $d = 2$  gælder  $\frac{OPT_g}{OPT} = 2$ .*

## Eksempel (Blokeret ring)



# Egenskaber

## Sætning

*p guillotinesnitbar hvis og kun hvis der ikke findes en blokeret ring i p.*

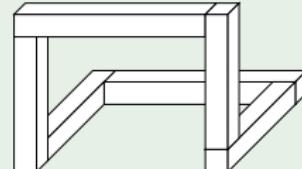
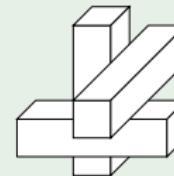
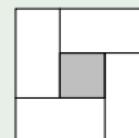
## Sætning

For  $d = 2$  gælder  $\frac{OPT_g}{OPT} = 2$ .

## Sætning

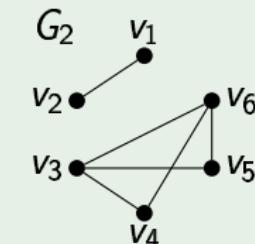
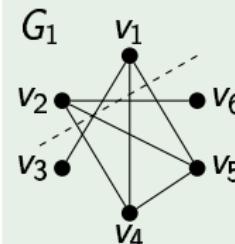
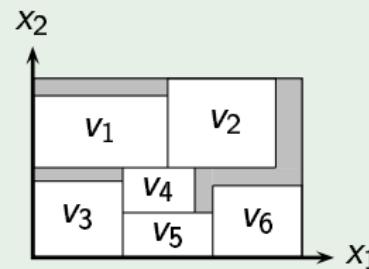
OPP-d med guillotinekrav er NP-hårdt.

## Eksempel (Blokeret ring)



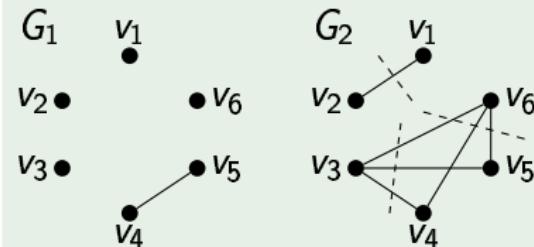
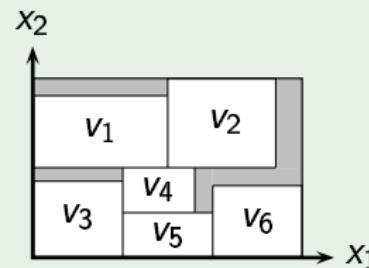
# Opløsning

## Eksempel



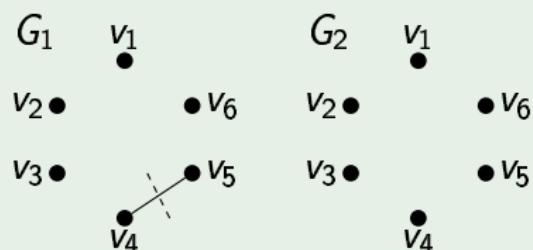
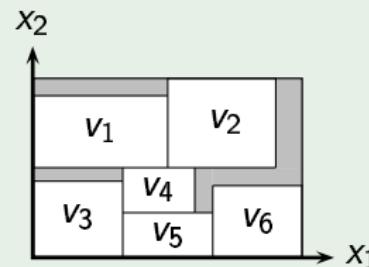
# Opløsning

## Eksempel



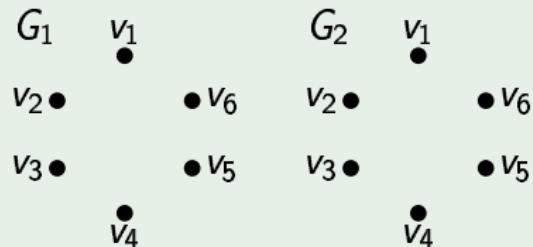
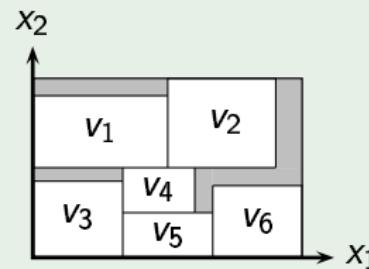
# Opløsning

## Eksempel



# Opløsning

## Eksempel

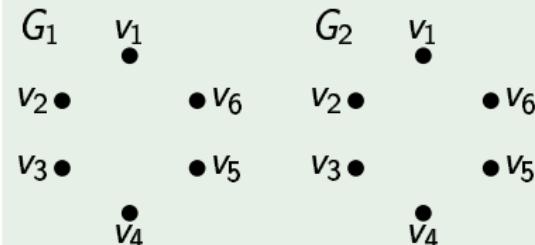
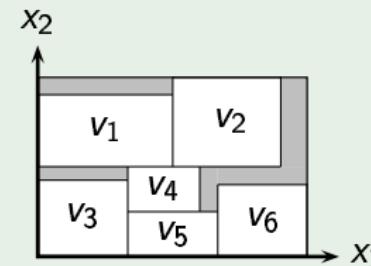


# Opløsning

Lad  $\alpha_1, \dots, \alpha_k$  angive normaler til snitplaner.

- For cutslice  $U$  i stage  $j$  er cutslice'ne for stage  $j + 1$  netop de sammenhængende komponenter i  $G_{\alpha_{j+1}}$ .

## Eksempel

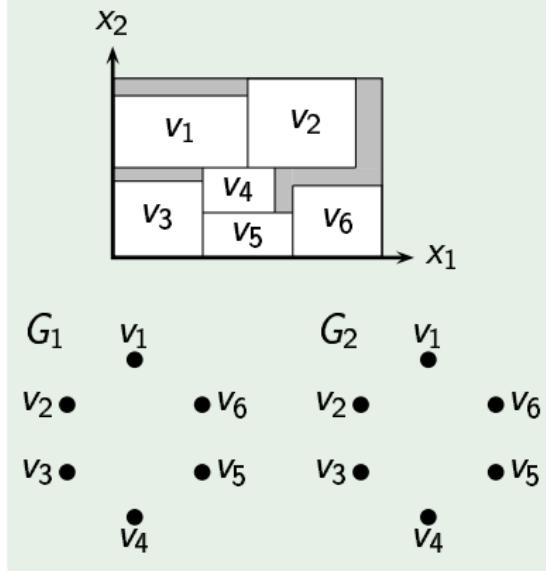


# Opløsning

Lad  $\alpha_1, \dots, \alpha_k$  angive normaler til snitplaner.

- For cutslice  $U$  i stage  $j$  er cutslice'ne for stage  $j + 1$  netop de sammenhængende komponenter i  $G_{\alpha_{j+1}}$ .
- $|U| > 1$  og  $|G_{\alpha_j}[U]| = 1$  for alle  $j \in \{1, \dots, d\} \Rightarrow p$  ej guillotinepakning.

## Eksempel

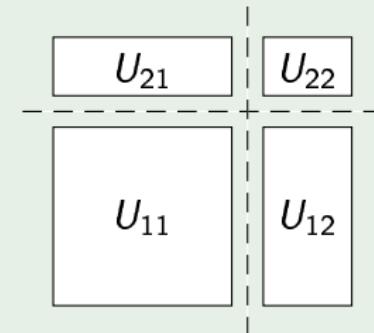


# Opløsning

Lad  $\alpha_1, \dots, \alpha_k$  angive normaler til snitplaner.

- For cutslice  $U$  i stage  $j$  er cutslice'ne for stage  $j + 1$  netop de sammenhængende komponenter i  $G_{\alpha_{j+1}}$ .
- $|U| > 1$  og  $|G_{\alpha_j}[U]| = 1$  for alle  $j \in \{1, \dots, d\} \Rightarrow p$  ej guillotinepakning.
- $|G_{\alpha_i}[U]| > 1$  og  $|G_{\alpha_j}[U]| > 1$  for  $i \neq j \Rightarrow \alpha_i$  og  $\alpha_j$  kan ombyttes.

## Eksempel



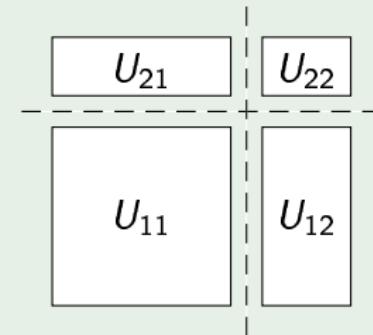
# Opløsning

Lad  $\alpha_1, \dots, \alpha_k$  angive normaler til snitplaner.

- For cutslice  $U$  i stage  $j$  er cutslice'ne for stage  $j + 1$  netop de sammenhængende komponenter i  $G_{\alpha_{j+1}}$ .
- $|U| > 1$  og  $|G_{\alpha_j}[U]| = 1$  for alle  $j \in \{1, \dots, d\} \Rightarrow p$  ej guillotinepakning.
- $|G_{\alpha_i}[U]| > 1$  og  $|G_{\alpha_j}[U]| > 1$  for  $i \neq j \Rightarrow \alpha_i$  og  $\alpha_j$  kan ombyttes.

Algoritme 2: Tilføj rekursivt guillotinetjek til algoritme 1.

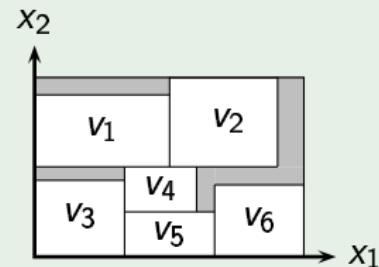
## Eksempel



# Måling af $x_i$ -bredde

## Eksempel

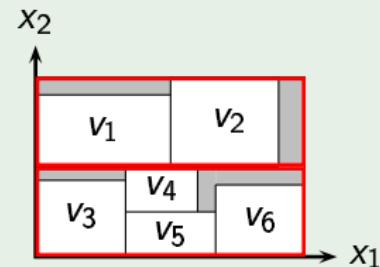
Mål  $w_2(V)$ :



# Måling af $x_i$ -bredde

## Eksempel

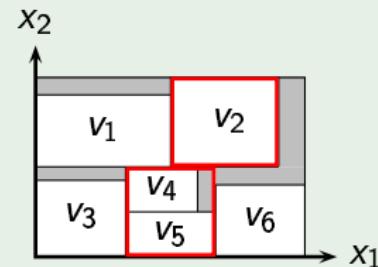
Mål  $w_2(V)$ :



# Måling af $x_i$ -bredde

## Eksempel

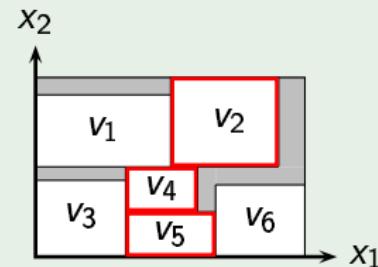
Mål  $w_2(V)$ :



# Måling af $x_i$ -bredde

## Eksempel

Mål  $w_2(V)$ :



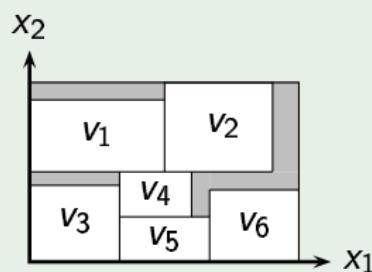
# Måling af $x_i$ -bredde

- Generelt udtryk for  $w_i(V)$ :

$$w_i(U) = \begin{cases} \sum_{U' \in G_{\alpha_j}[U]} w_i(U') & , \alpha_j = i \\ \max_{U' \in G_{\alpha_j}[U]} w_i(U') & , \alpha_j \neq i \end{cases}$$

## Eksempel

Mål  $w_2(V)$ :



# Måling af $x_i$ -bredde

- Generelt udtryk for  $w_i(V)$ :

$$w_i(U) = \begin{cases} \sum_{U' \in G_{\alpha_j}[U]} w_i(U') & , \alpha_j = i \\ \max_{U' \in G_{\alpha_j}[U]} w_i(U') & , \alpha_j \neq i \end{cases}$$

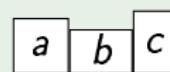
- Måling ikke entydig: Afhænger af snitrækkefølge  $\alpha_1, \dots, \alpha_k$ .

## Eksempel

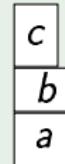
$G_1$  •a       $G_2$  •a

b•      •c      b•      •c

Snitretning:  $x_1 x_2$



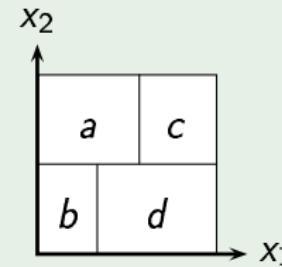
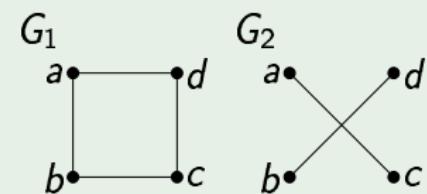
Snitretning:  $x_2 x_1$



## P1, P2 og P3

P1 Ikke nødvendig hvis  $G_1, \dots, G_d$  er opløselige.

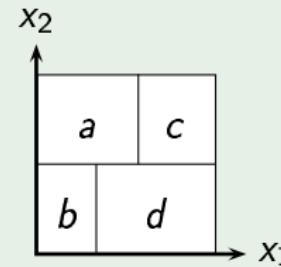
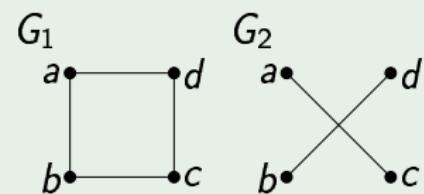
## Eksempel



## P1, P2 og P3

- P1 Ikke nødvendig hvis  $G_1, \dots, G_d$  er opløselige.
- P2  $x_i$ -bredde måles ved opløsning.

## Eksempel



# P1, P2 og P3

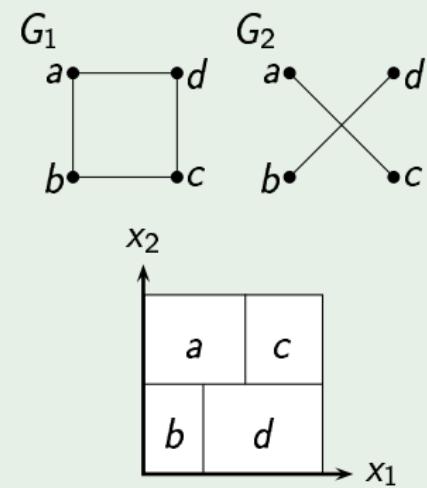
P1 Ikke nødvendig hvis  $G_1, \dots, G_d$  er opløselige.

P2  $x_i$ -bredde måles ved oplosning.

P3 Automatisk opfyldt hvis  $p$  er opløselig.

Betrægt dog  $\bigcap_{i=1}^d \mathcal{E}_{+,i}$  for at sikre opløselighed.

## Eksempel



# P1, P2 og P3

P1 Ikke nødvendig hvis  $G_1, \dots, G_d$  er opløselige.

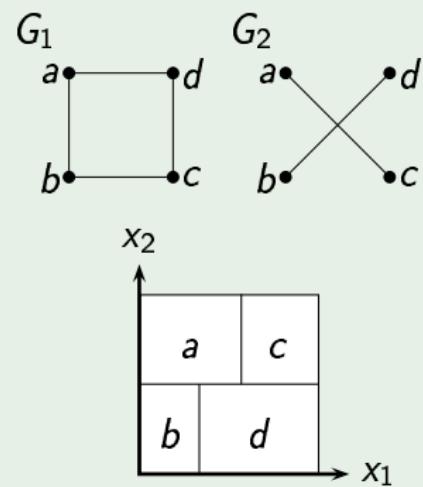
P2  $x_i$ -bredde måles ved oplosning.

P3 Automatisk opfyldt hvis  $p$  er opløselig.

Betrægt dog  $\bigcap_{i=1}^d \mathcal{E}_{+,i}$  for at sikre opløselighed.

Algoritme 3: Drop P1-tjek og brug kun oplosningsbaseret måling.

## Eksempel



# Fordeler/ulemper

$M_C$ : Konventionel klikbaseret måling

$M_D$ : Opløsningsbaseret måling

	$M_C$	$M_D$
Redundans	+	-

## Eksempel

$G_1$  •a

b•     •c

$G_2$  •a

b•     •c

# Fordeler/ulemper

$M_C$ : Konventionel klikbaseret måling

$M_D$ : Opløsningsbaseret måling

	$M_C$	$M_D$
Redundans	+	-
Tid pr. iteration	-	+

## Eksempel

$G_1$  •a

$b$ •    •c

$G_2$  •a

$b$ •    •c

$G_1$  •a

$b$ •    •c

$G_2$  •a

$b$ •    •c

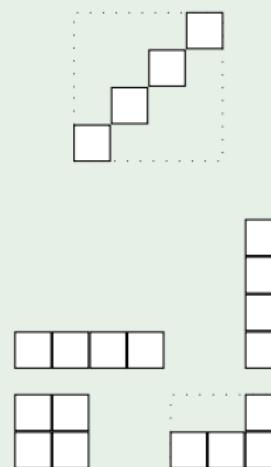
# Fordeler/ulemper

$M_C$ : Konventionel klikbaseret måling

$M_D$ : Opløsningsbaseret måling

	$M_C$	$M_D$
Redundans	+	-
Tid pr. iteration	-	+
Altid minimal	-	+

## Eksempel



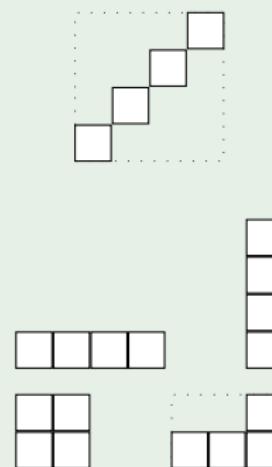
# Fordeler/ulemper

$M_C$ : Konventionel klikbaseret måling

$M_D$ : Opløsningsbaseret måling

	$M_C$	$M_D$
Redundans	+	-
Tid pr. iteration	-	+
Altid minimal	-	+
Mindst søgerum	?	?

## Eksempel



# Fordeler/ulemper

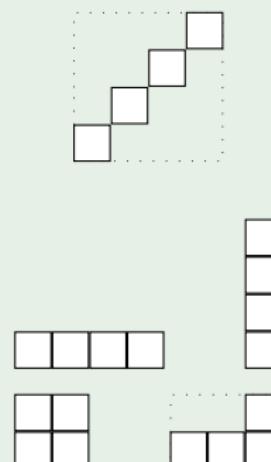
$M_C$ : Konventionel klikbaseret måling

$M_D$ : Opløsningsbaseret måling

	$M_C$	$M_D$
Redundans	+	-
Tid pr. iteration	-	+
Altid minimal	-	+
Mindst søgerum	?	?

$M_D$  baseret algoritme 3 implementeret.

## Eksempel

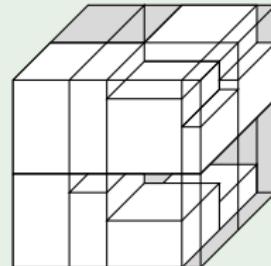
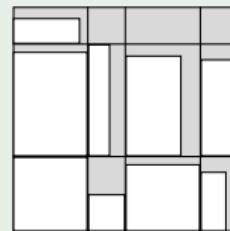


# Disposition

- 1 Problemstillinger
- 2 Fekete og Schepers' framework
- 3 Guillotinesnit
- 4 Sticky cutting
- 5 Resultater
- 6 Afrunding

# Stickycutting / sætterkassepakning

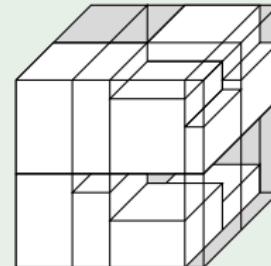
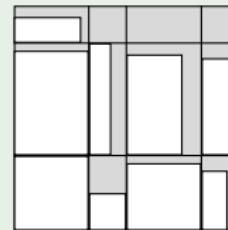
## Eksempel



# Stickycutting / sætterkassepakning

- Alle snit deler original container.

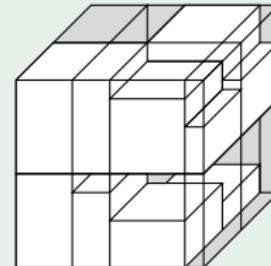
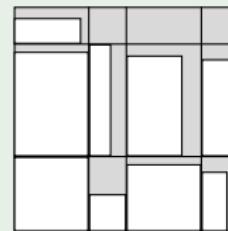
## Eksempel



# Stickycutting / sætterkassepakning

- Alle snit deler original container.
- $k = d$ .

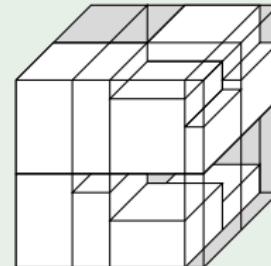
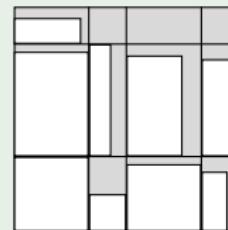
## Eksempel



# Stickycutting / sætterkassepakning

- Alle snit deler original container.
- $k = d$ .
- Svær at regne på: Må betragte hele pakningen.

## Eksempel



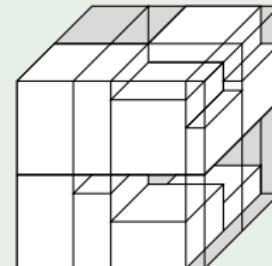
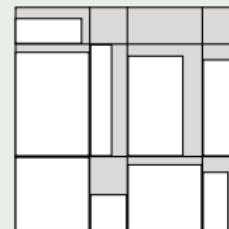
# Stickycutting / sætterkassepakning

- Alle snit deler original container.
- $k = d$ .
- Svær at regne på: Må betragte hele pakningen.

## Formodning

*OPP-d med stickycuttingkrav er NP-hårdt.*

## Eksempel



# Stickycutting / sætterkassepakning

- Alle snit deler original container.
- $k = d$ .
- Svær at regne på: Må betragte hele pakningen.

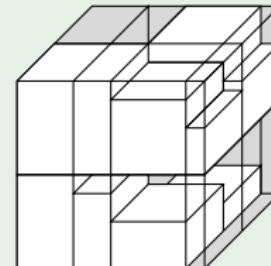
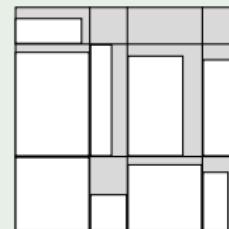
## Formodning

*OPP-d med stickycuttingkrav er NP-hårdt.*

## Formodning

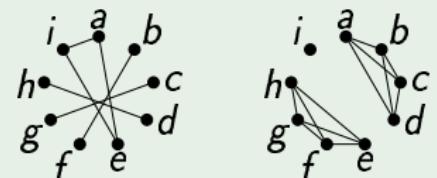
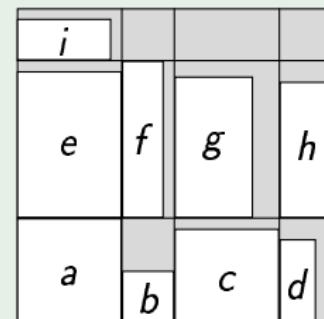
For  $d = 2$  er  $\frac{OPT_s}{OPT_g} = 2$  og  $\frac{OPT_s}{OPT} = 4$ .

## Eksempel



# Grafrepresentation

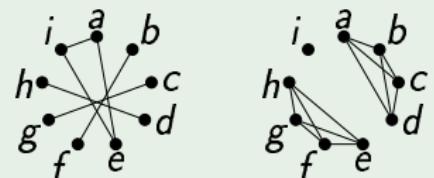
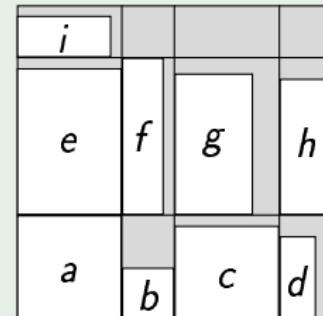
## Eksempel



# Grafrepresentation

- Alle sammenhængende komponenter i  $G_i$  er kliker.

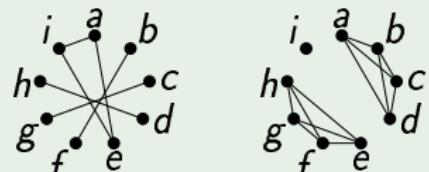
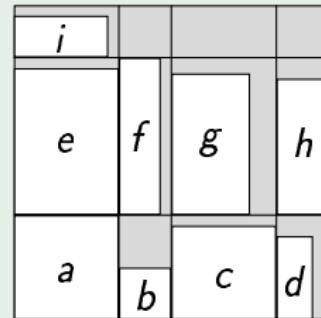
## Eksempel



# Grafrepresentation

- Alle sammenhængende komponenter i  $G_i$  er klier.
- Fix  $e = (u, v) \Rightarrow$  fix også  $e' = (u', v')$  for  $u' \in \mathcal{C}_u$  og  $v' \in \mathcal{C}_v$ .

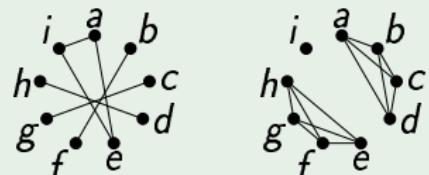
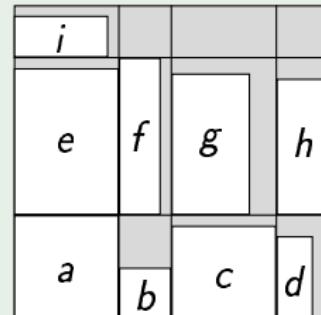
## Eksempel



# Grafrepresentation

- Alle sammenhængende komponenter i  $G_i$  er kliker.
- Fix  $e = (u, v) \Rightarrow$  fix også  $e' = (u', v')$  for  $u' \in \mathcal{C}_u$  og  $v' \in \mathcal{C}_v$ .
- Fix kanter  $\rightarrow$  merge kliker.

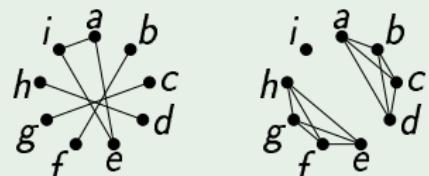
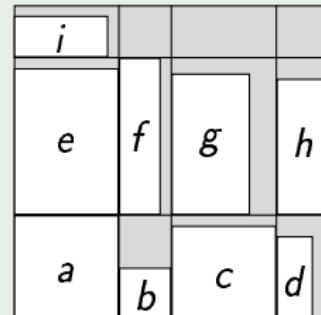
## Eksempel



# Grafrepresentation

- Alle sammenhængende komponenter i  $G_i$  er klier.
- Fix  $e = (u, v) \Rightarrow$  fix også  $e' = (u', v')$  for  $u' \in \mathcal{C}_u$  og  $v' \in \mathcal{C}_v$ .
- Fix kanter  $\rightarrow$  merge klier.
- $w_i(V) = \sum_{\mathcal{C} \in G_i} \max_{v \in \mathcal{C}} w_i(v)$

## Eksempel



# P1, P2 og P3

P1 Ingen kordeløs  $C_4$  i  $G_i$ :

# P1, P2 og P3

P1 Ingen kordeløs  $C_4$  i  $G_i$ : ✓

Ingen 2-kordeløs ulige kreds i  $G_i^C$ :

# P1, P2 og P3

- P1 Ingen kordeløs  $C_4$  i  $G_i$ : ✓  
Ingen 2-kordeløs ulige kreds i  $G_i^C$ : ✓  
Automatisk opfyldt.

# P1, P2 og P3

- P1 Ingen kordeløs  $C_4$  i  $G_i$ : ✓  
Ingen 2-kordeløs ulige kreds i  $G_i^C$ : ✓  
Automatisk opfyldt.
- P2 Tjekkes i  $O(|V|)$  tid.

# P1, P2 og P3

- P1 Ingen kordeløs  $C_4$  i  $G_i$ : ✓  
Ingen 2-kordeløs ulige kreds i  $G_i^C$ : ✓  
Automatisk opfyldt.
- P2 Tjekkes i  $O(|V|)$  tid.
- P3 Betragt  $\bigcap_{i=1}^d \mathcal{E}_{+,i}$  som i algoritme 1.

# Fordeler/ulemper

- + Grafteoretisk meget pæn.

# Fordeler/ulemper

- + Grafteoretisk meget pæn.
- + Mindre søgerum end for guillotinepakninger.

# Fordeler/ulemper

- + Grafteoretisk meget pæn.
- + Mindre søgerum end for guillotinepakninger.
- + Kan afskære søgerum yderligere med volumetjek.

# Fordeler/ulemper

- + Grafteoretisk meget pæn.
- + Mindre søgerum end for guillotinepakninger.
- + Kan afskære søgerum yderligere med volumetjek.
- + Flere iterationer pr. sekund.

# Fordeler/ulemper

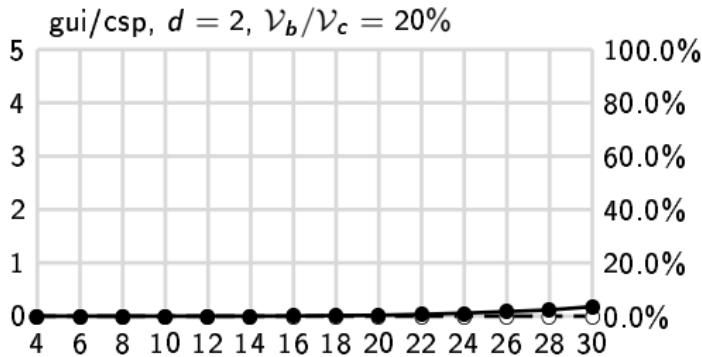
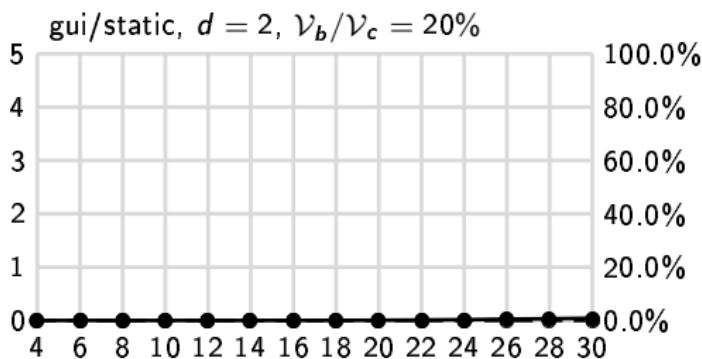
- + Grafteoretisk meget pæn.
- + Mindre søgerum end for guillotinepakninger.
- + Kan afskære søgerum yderligere med volumetjek.
- + Flere iterationer pr. sekund.
- Stor redundans i søgerum.

# Disposition

- 1 Problemstillinger
- 2 Fekete og Schepers' framework
- 3 Guillotinesnit
- 4 Sticky cutting
- 5 Resultater
- 6 Afrunding

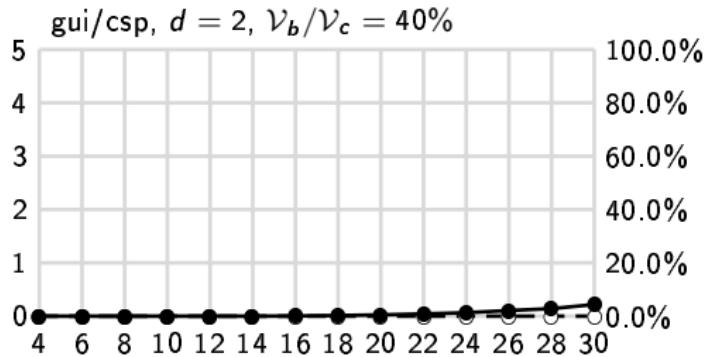
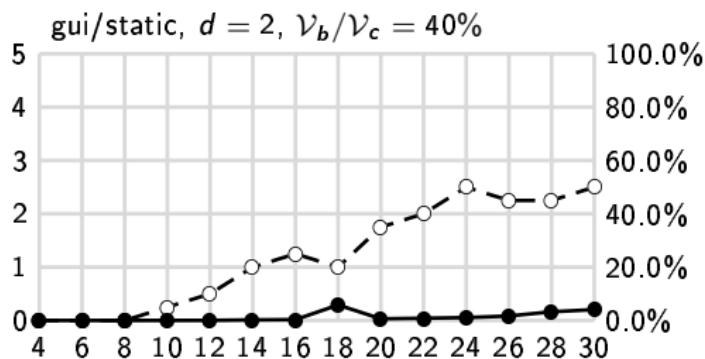
# Guillotine: F&S vs. CSP-løsner (Pisinger) for $d = 2$

- 20 feasible random-instanser pr. punkt.
- Timeout: 6 sekunder.
- CSP har højere grænse for divergens.



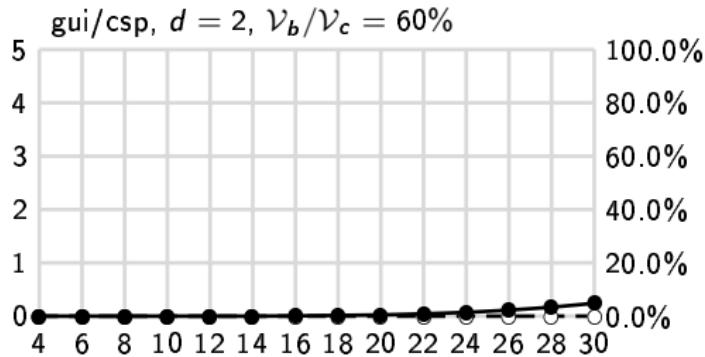
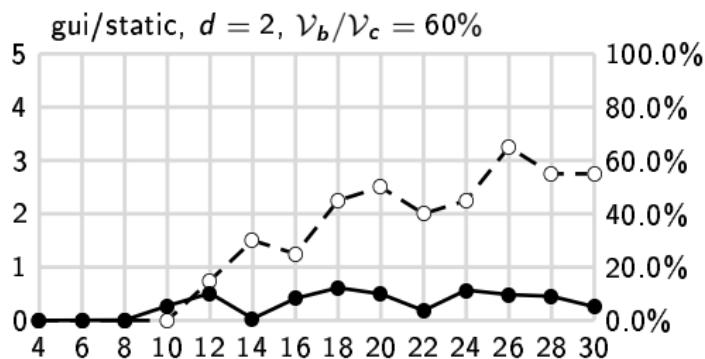
# Guillotine: F&S vs. CSP-løsner (Pisinger) for $d = 2$

- 20 feasible random-instanser pr. punkt.
- Timeout: 6 sekunder.
- CSP har højere grænse for divergens.



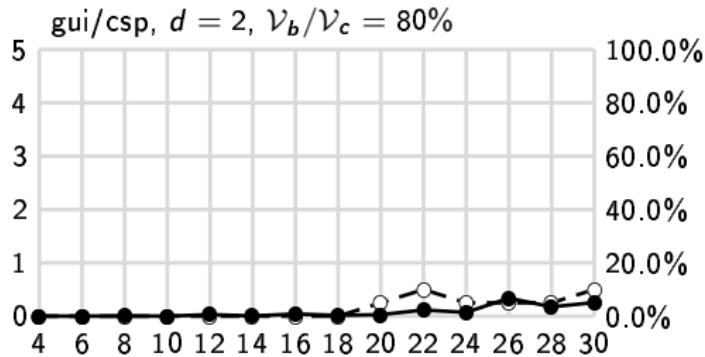
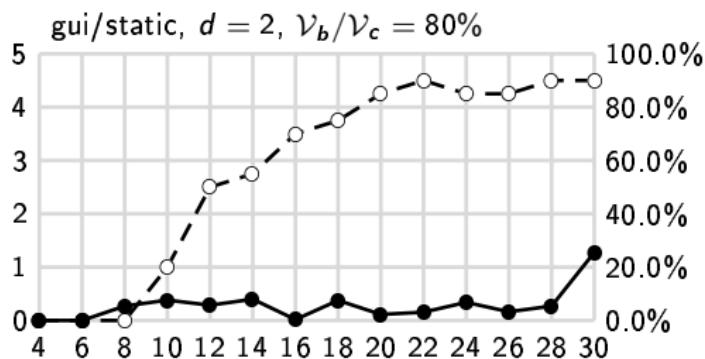
# Guillotine: F&S vs. CSP-løsner (Pisinger) for $d = 2$

- 20 feasible random-instanser pr. punkt.
- Timeout: 6 sekunder.
- CSP har højere grænse for divergens.



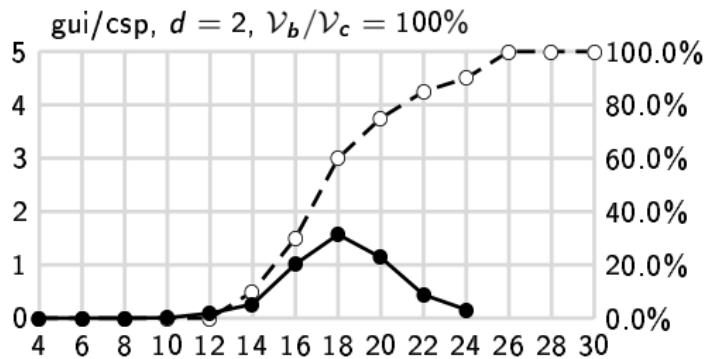
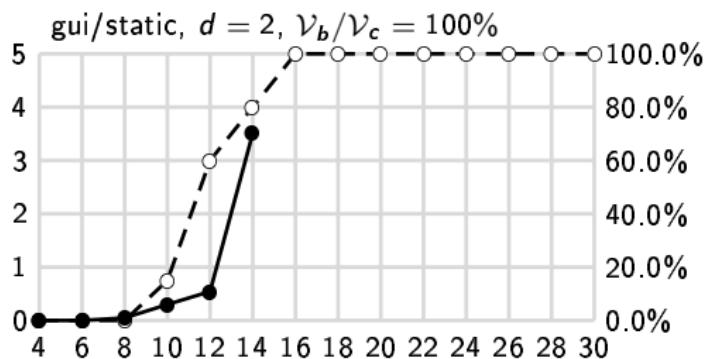
# Guillotine: F&S vs. CSP-løser (Pisinger) for $d = 2$

- 20 feasible random-instanser pr. punkt.
- Timeout: 6 sekunder.
- CSP har højere grænse for divergens.



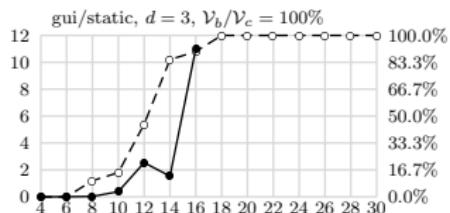
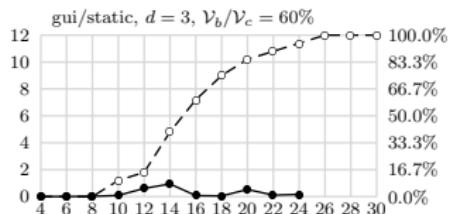
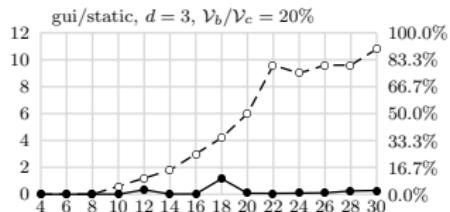
# Guillotine: F&S vs. CSP-løsner (Pisinger) for $d = 2$

- 20 feasible random-instanser pr. punkt.
- Timeout: 6 sekunder.
- CSP har højere grænse for divergens.

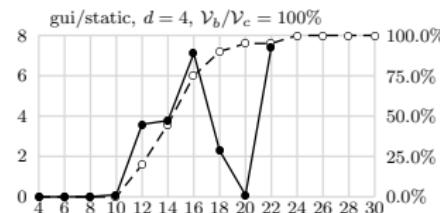
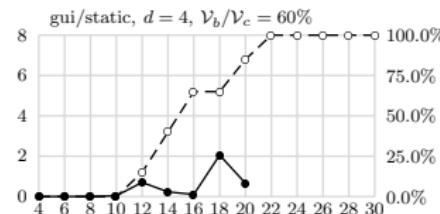
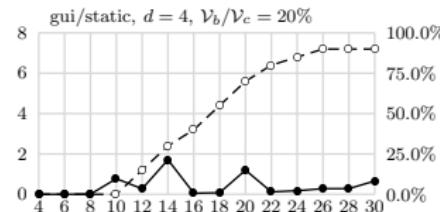


# F&S Guillotine-løsner for $d = 3$ og $d = 4$

$d = 3$ , timeout: 20 s.

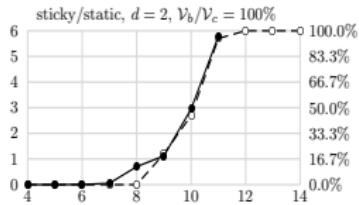
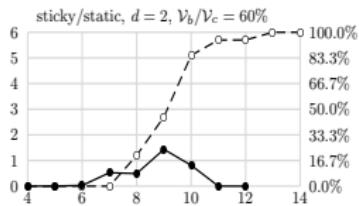
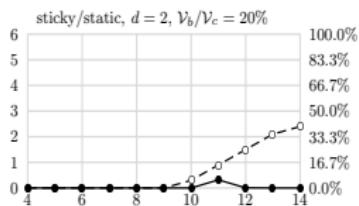


$d = 4$ , timeout: 30 s.

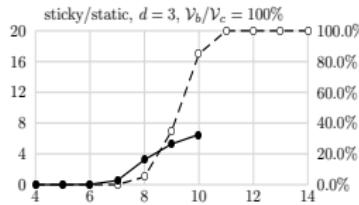
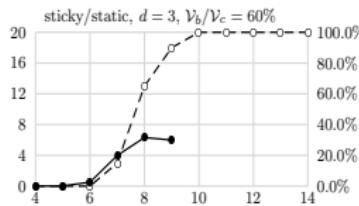
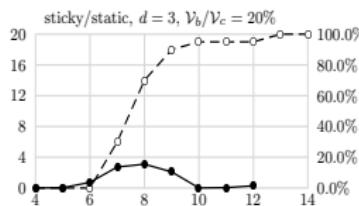


# Stickycutting-løsner for $d = 2, 3, 4$

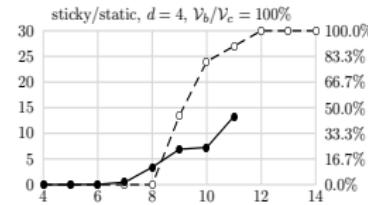
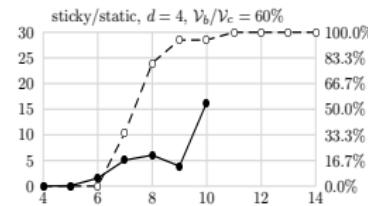
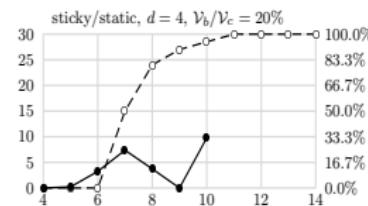
$d = 2$ , timeout: 6 s.



$d = 3$ , timeout: 20 s.



$d = 4$ , timeout: 30 s.



# Disposition

- 1 Problemstillinger
- 2 Fekete og Schepers' framework
- 3 Guillotinesnit
- 4 Sticky cutting
- 5 Resultater
- 6 Afrunding

# Generelt

- Optimér implementation. gprof profiling:

Delroutine	Sek. stl::hash	Sek. statistik
packingclass_test()	215	36
update_searchinfo()	264	32

# Generelt

- Optimér implementation. gprof profiling:

Delroutine	Sek. stl::hash	Sek. statistik
packingclass_test()	215	36
update_searchinfo()	264	32

- Benyt conservative scales (bevarer feasibility)

# Algoritmiske overvejelser

Sticky cutting-løser:

- Knuder pr. sekund: 1100 (gui) vs. 5200 (sticky)

Guillotine-løser:

# Algoritmiske overvejelser

Stickycutting-løser:

- Knuder pr. sekund: 1100 (gui) vs. 5200 (sticky)
- Problem: Mange måder at konstruere samme klike

Guillotine-løser:

# Algoritmiske overvejelser

Sticky cutting-løsere:

- Knuder pr. sekund: 1100 (gui) vs. 5200 (sticky)
- Problem: Mange måder at konstruere samme klike
- Er dog hurtig til at erklære infeasible:

	Guillotine		Sticky	
	Tid	Knuder	Tid	Knuder
2dinf7	2.93	9673	0.35	3301
2dinf8	61.80	192961	1.93	16547

Guillotine-løsere:

# Algoritmiske overvejelser

## Sticky cutting-løsere:

- Knuder pr. sekund: 1100 (gui) vs. 5200 (sticky)
- Problem: Mange måder at konstruere samme klike
- Er dog hurtig til at erklære infeasible:

	Guillotine		Sticky	
	Tid	Knuder	Tid	Knuder
2dinf7	2.93	9673	0.35	3301
2dinf8	61.80	192961	1.93	16547

## Guillotine-løsere:

- Stor redundans med  $M_D$ -topologier → tidlig divergens

# Algoritmiske overvejelser

## Sticky cutting-løsere:

- Knuder pr. sekund: 1100 (gui) vs. 5200 (sticky)
- Problem: Mange måder at konstruere samme klike
- Er dog hurtig til at erklære infeasible:

	Guillotine		Sticky	
	Tid	Knuder	Tid	Knuder
2dinf7	2.93	9673	0.35	3301
2dinf8	61.80	192961	1.93	16547

## Guillotine-løsere:

- Stor redundans med  $M_D$ -topologier → tidlig divergens
- Genindfør  $M_C$ -topologier (kræver P1)

# Bidrag

# Bidrag

- Fekete & Schepers' algoritme udvidet til guillotinepakninger.  
To algoritmer præsenteret.

# Bidrag

- Fekete & Schepers' algoritme udvidet til guillotinepakninger.  
To algoritmer præsenteret.
- Bevis for  $\frac{\text{OPT}_g}{\text{OPT}} = 2$  for  $d = 2$  givet i samarbejde med David Pisinger.

# Bidrag

- Fekete & Schepers' algoritme udvidet til guillotinepakninger.  
To algoritmer præsenteret.
- Bevis for  $\frac{\text{OPT}_g}{\text{OPT}} = 2$  for  $d = 2$  givet i samarbejde med David Pisinger.
- Introduktion af sticky cuttings med volumetjek og formodning om  $\frac{\text{OPT}_s}{\text{OPT}_g} = 2$  og  $\frac{\text{OPT}_s}{\text{OPT}} = 4$  for  $d = 2$ .  
Teorien fra sticky cuttings kan bruges for konventionelle guillotinepakninger for tilfældet  $k = d = 2$ .

# Bidrag

- Fekete & Schepers' algoritme udvidet til guillotinepakninger.  
To algoritmer præsenteret.
- Bevis for  $\frac{\text{OPT}_g}{\text{OPT}} = 2$  for  $d = 2$  givet i samarbejde med David Pisinger.
- Introduktion af stickycuttings med volumetjek og formodning om  $\frac{\text{OPT}_s}{\text{OPT}_g} = 2$  og  $\frac{\text{OPT}_s}{\text{OPT}} = 4$  for  $d = 2$ .  
Teorien fra stickycuttings kan bruges for konventionelle guillotinepakninger for tilfældet  $k = d = 2$ .
- Implementation af sticky cutting- og  $M_D$ -baserede løsere.  
Første løsere for vilkårlige  $d$ .

# Bidrag

- Fekete & Schepers' algoritme udvidet til guillotinepakninger.  
To algoritmer præsenteret.
- Bevis for  $\frac{\text{OPT}_g}{\text{OPT}} = 2$  for  $d = 2$  givet i samarbejde med David Pisinger.
- Introduktion af stickycuttings med volumetjek og formodning om  $\frac{\text{OPT}_s}{\text{OPT}_g} = 2$  og  $\frac{\text{OPT}_s}{\text{OPT}} = 4$  for  $d = 2$ .  
Teorien fra stickycuttings kan bruges for konventionelle guillotinepakninger for tilfældet  $k = d = 2$ .
- Implementation af sticky cutting- og  $M_D$ -baserede løsere.  
Første løsere for vilkårlige  $d$ .
- Introduktion af pakningstræer med brute-force algoritme.