Justification de l'Optimisation FloydWarshall opérant avec un seul tableau

Nicolas RAPIN
CEA LIST
nicolas.rapin@cea.fr
Pour PC d'Algorithmique, Ecole Centrale
2016

Pseudo-code [modifier | modifier le code]

Donnons d'abord une première version d'après l'analyse faite dans la section précédente.Le pseudo-code suivant effectue ce calcul :

```
fonction FloydWarshall (G)
  W^0 := matrice d'adjacence de G (matrice n × n)
  for k := 1 to n
      for i := 1 to n
          for j := 1 to n
              W^k_ij = min(W^(k-1)_ij, W^(k-1)_ik + W^(k-1)_kj)
  retourner W^n
```

On peut optimiser l'algorithme en effectuant le calcul en place dans une unique matrice $\mathcal{W}$. Le pseudo-code suivant effectue ce calcul :

```
fonction FloydWarshall (G)
  W :=matrice d'adjacence de G (matrice n × n)
  for k := 1 to n
      for i := 1 to n
          for j := 1 to n
              W_ij := min(W_ij, W_ik + W_kj)
  retourner W
```

Les n+1 sommets sont représentés par les entiers [0,n].

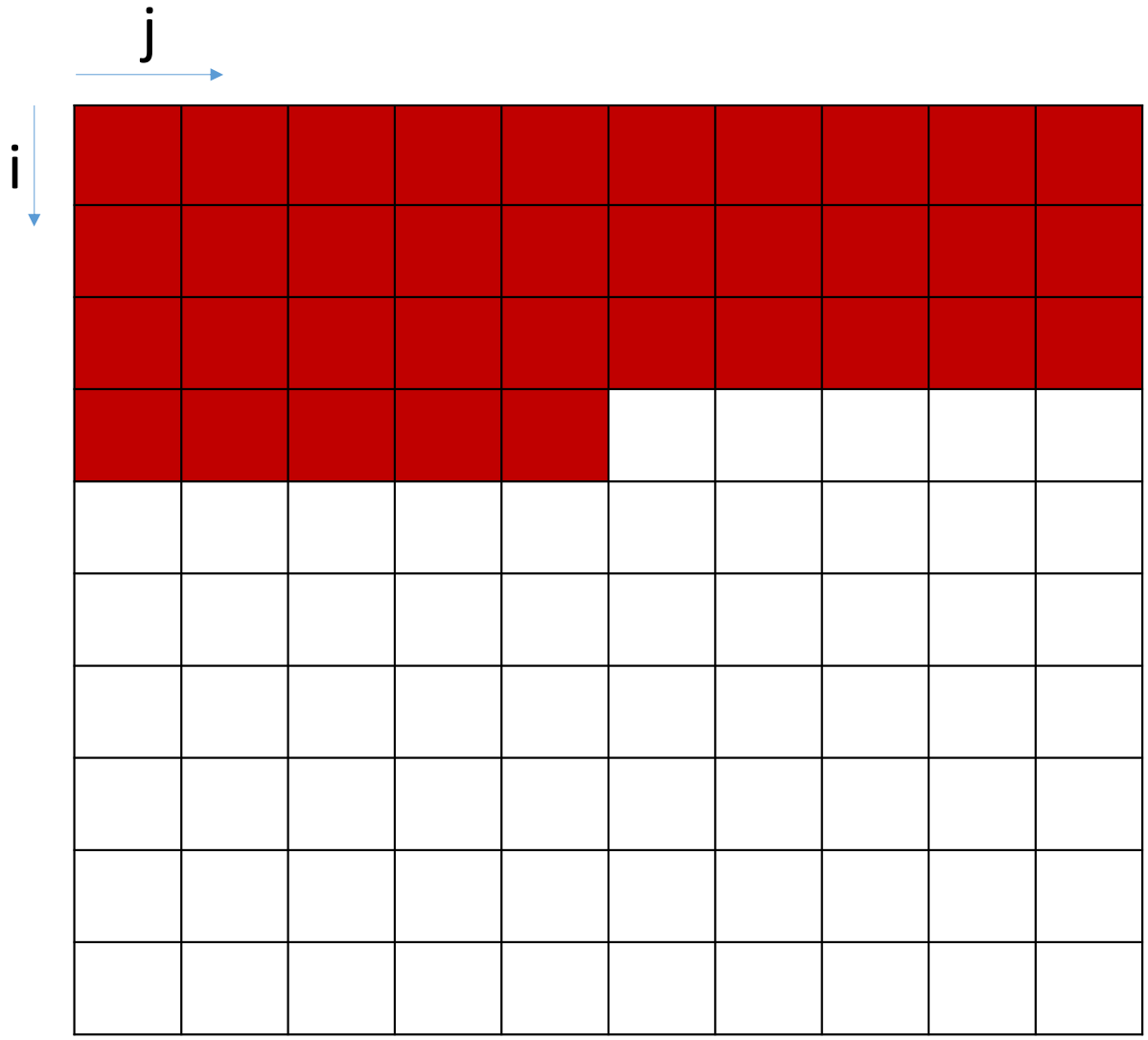
En posant que $D^k_{i,j}$ représente la distance du nœud i au nœud j à l'itération k le raisonnement récursif de FloydWarshall aboutit à:

$$D^k_{i,j} = \min (D^{k-1}_{i,j} , D^{k-1}_{i,k} + D^{k-1}_{k,j})$$

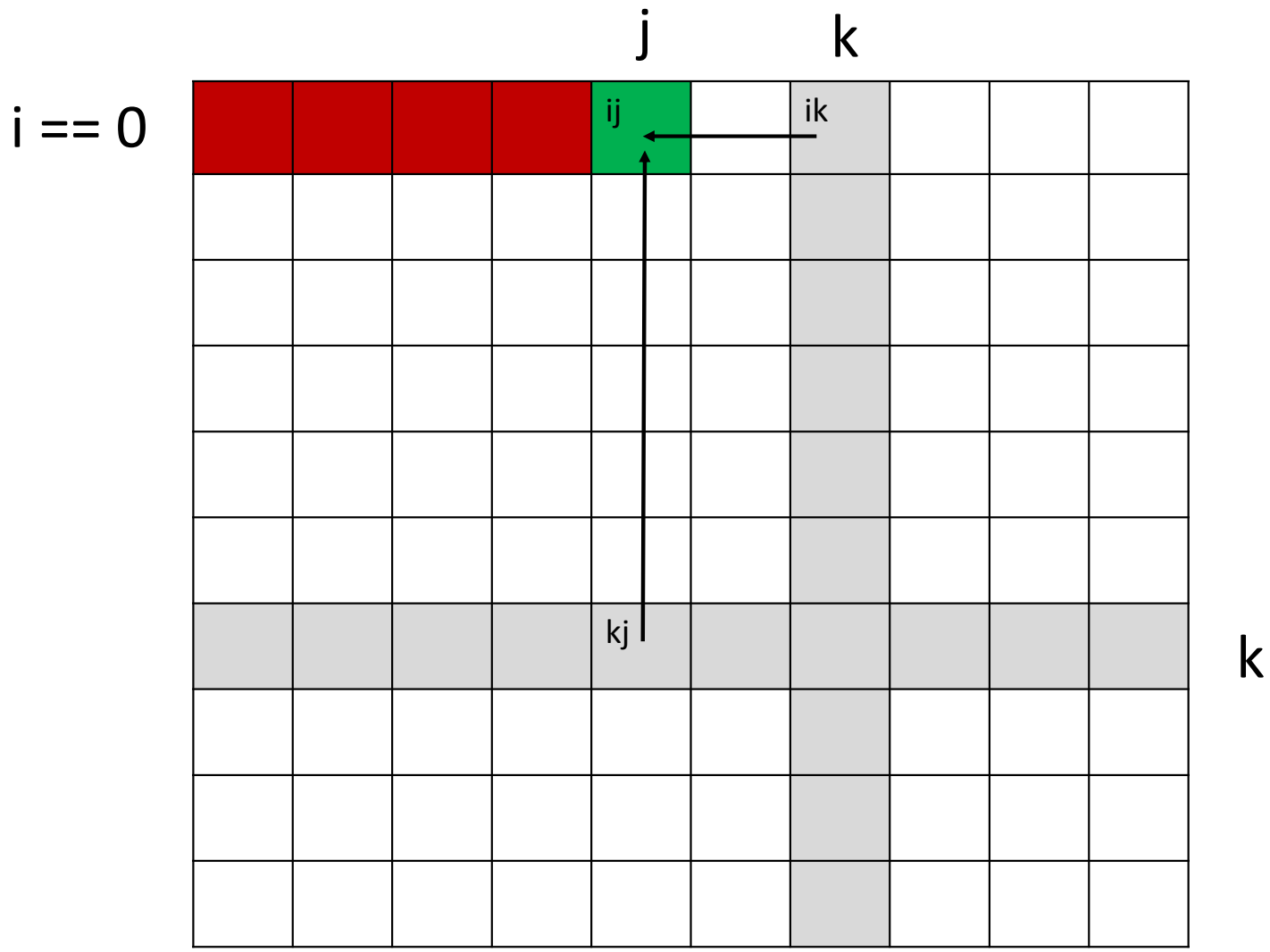
Question: peut-on faire ce calcul avec une seule matrice plutôt qu'avec deux matrices (une pour k-1 et une pour k) ?

Autrement dit, si D est un tableau informatique peut-on faire simplement:

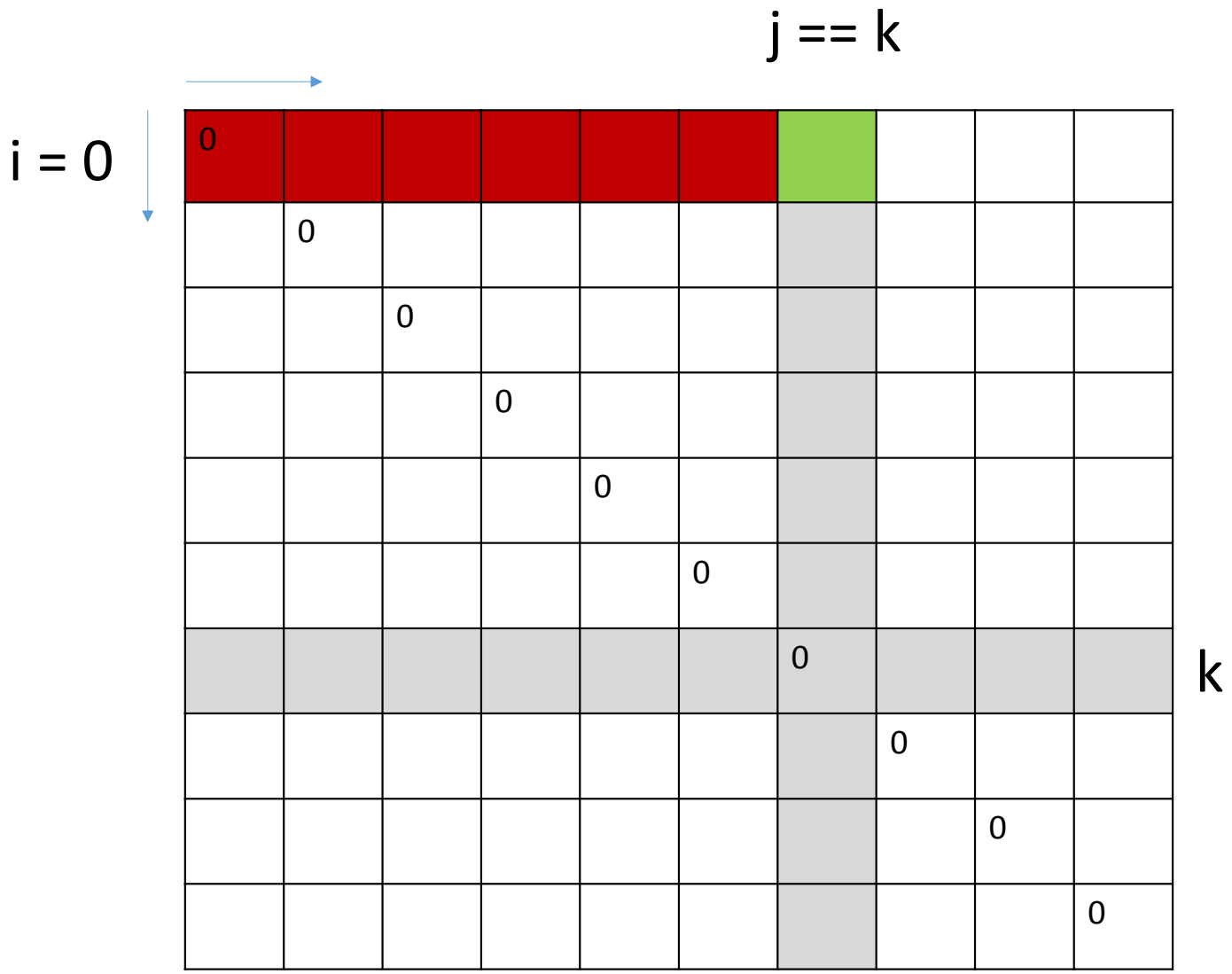
$$D_{i,j} = \min (D_{i,j} , D_{i,k} + D_{k,j})$$



Les boucles en i,j balayent la matrice comme représenté.



Dans les « cas initiaux » (i == 0) la définition de $D^k_{i,j}$ dépend bien de valeurs définies à l'itération k-1.



$D_{i,j} = \min (D_{i,j} , D_{i,k} + D_{k,j})$

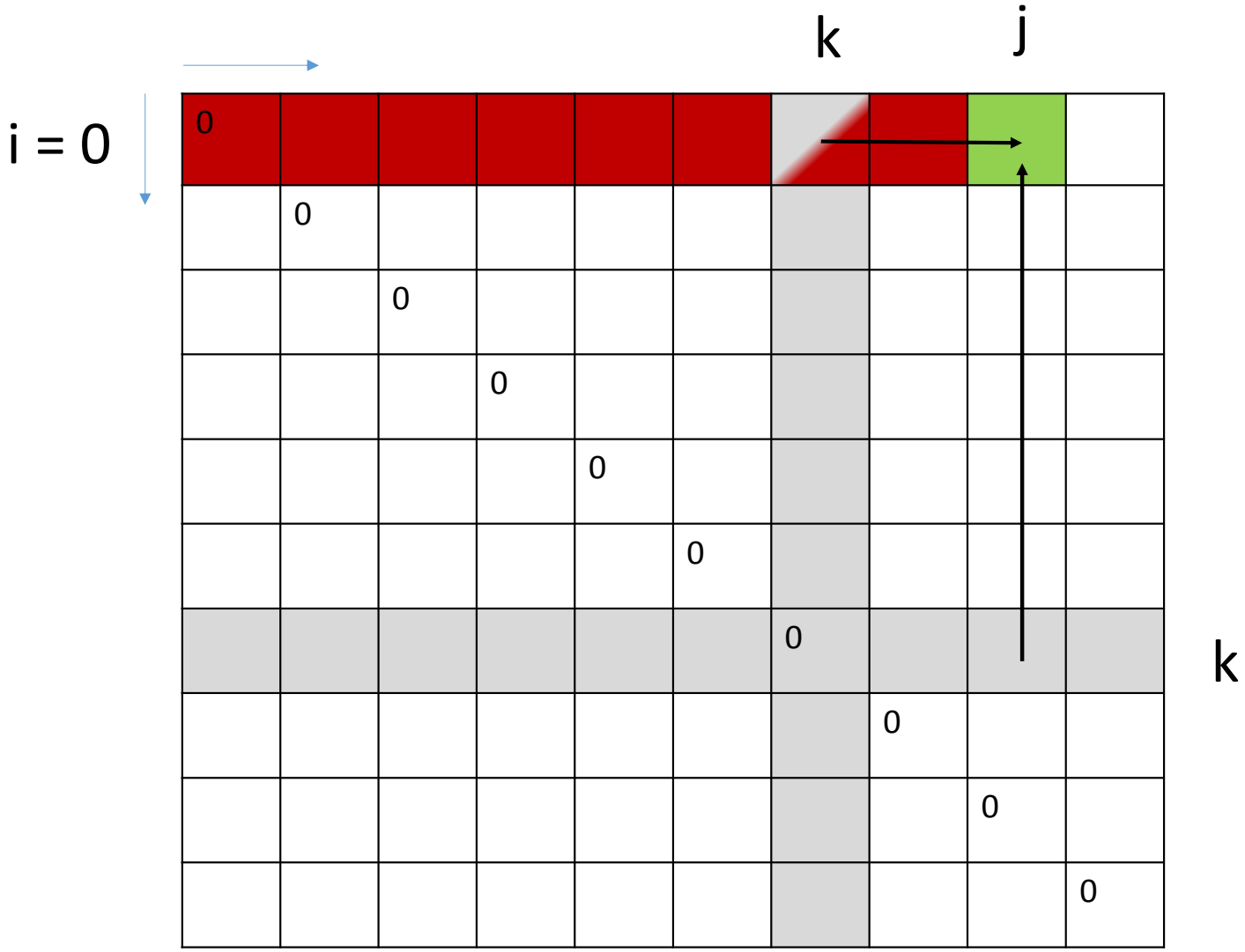
$D_{i,j} = \min (D_{i,j} , D_{i,j} + D_{k,k})$

$D_{i,j} = \min (D_{i,j} , D_{i,j} + 0)$

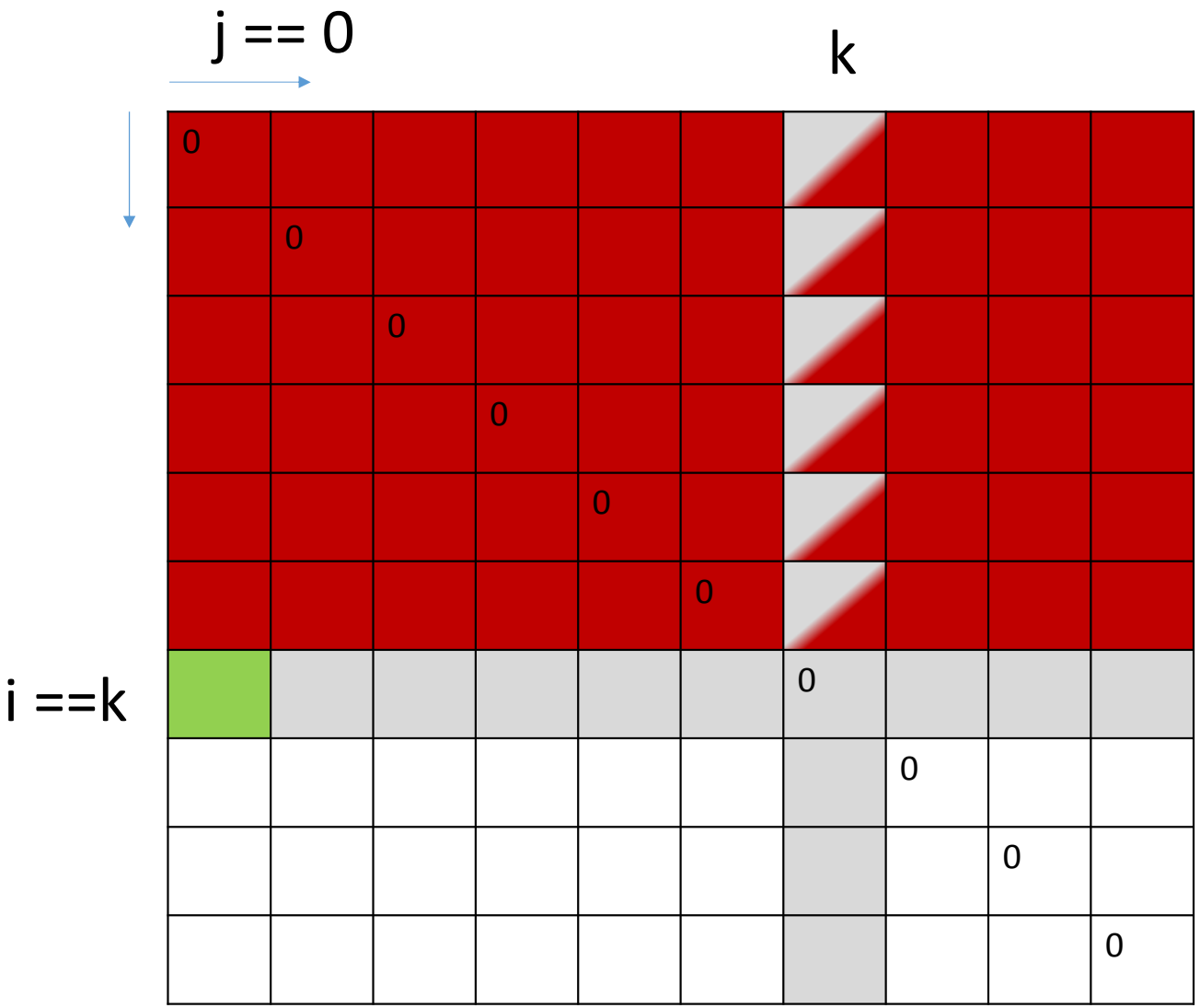
$D_{i,j} = \min (D_{i,j} , D_{i,j})$

$D_{i,j} = D_{i,j}$

$D_{i,j}$ est inchangé !



Puisque $D_{0,k}$ est resté inchangé (cf slide précédent) dans ce cas nous avons encore une définition de $D_{i,j}$ qui repose bien sur les valeurs à l'itération k-1.



$D_{i,j} = \min (D_{i,j} , D_{i,k} + D_{k,j})$

$D_{i,j} = \min (D_{i,j} , 0 + D_{i,j})$

$D_{i,j} = \min (D_{i,j} , D_{i,j})$

$D_{i,j} = D_{i,j}$

$D_{i,j}$ est inchangé !

Branch & Bound Illustration

Nicolas RAPIN
CEA LIST
nicolas.rapin@cea.fr
Pour PC d'Algorithmique, Ecole Centrale
2016

Nœud (problème) Initial

Minimiser $X_B+X_C+X_D+X_E$

Couv A	X_B	X_C	X_D		≥1
Couv C	X_B	X_C		X_E	≥1
Couv D	X_B		X_D	X_E	≥1
Couv F		X_C	X_D	X_E	≥1

Solveur linéaire en Réels:
$X_B == X_C == X_D == X_E = 1/3$
Cout: 4/3

$X_B == 0$

$X_B == 1$

Minimiser $X_C+X_D+X_E$

Couv A	**0**	X_C	X_D		≥1
Couv C	**0**	X_C		X_E	≥1
Couv D	**0**		X_D	X_E	≥1
Couv F		X_C	X_D	X_E	≥1

Solveur linéaire en Réels (X_B==0) :
$X_C == X_D == X_E == 1/2$
Cout: 3/2
(pas élagué car best_sol n'est pas définie)

Minimiser $X_C+X_D+X_E$

Couv A	**1**	X_C	X_D		≥1
Couv C	**1**	X_C		X_E	≥1
Couv D	**1**		X_D	X_E	≥1
Couv F		X_C	X_D	X_E	≥1

Solveur linéaire en Réels (X_B==1) :
$X_C == 1$, $X_D == X_E == 0$
Cout: 2
(tous entiers -> solution -> nouvelle best_sol
pas remis dans les nœuds actifs)

Choix du nœud à développer:

- Le plus profond
- (à profondeur égale) celui de coût moindre

Revient à placer les nœuds dans un tas min avec comme poids des couples (p,c), « p » pour profondeur, « c » pour coût, munis de l'ordre $\angle$ satisfaisant:

(p,c) $\angle$ (p',c') ssi une des conditions suivantes est satisfaite:

- p > p'
- p == p' et c < c'

Puis à défiler du tas les nœud par un pop().

Donc ici le pop() défile le nœud **orange** (qui est seul dans le tas)

Minimiser $X_C+X_D+X_E$

Couv A	**0**	X_C	X_D		≥1
Couv C	**0**	X_C		X_E	≥1
Couv D	**0**		X_D	X_E	≥1
Couv F		X_C	X_D	X_E	≥1

Solveur linéaire en Réels:
$X_C==X_D==X_E = 1/2$
Cout: 3/2

$X_C == 0$

$X_C == 1$

Minimiser X_D+X_E

Couv A		**0**	X_D		≥1
Couv C		**0**		X_E	≥1
Couv D			X_D	X_E	≥1
Couv F		**0**	X_D	X_E	≥1

Solveur linéaire en Réels ($X_B==0$, $X_C==0$):
$X_D== X_E == 1$
Cout: 2
(tous entiers -> solution -> pas meilleure que best_sol,
Pas mis dans les nœuds actifs)

Minimiser X_D+X_E

Couv A		**1**	X_D		≥1
Couv C		**1**		X_E	≥1
Couv D			X_D	X_E	≥1
Couv F		**1**	X_D	X_E	≥1

Solveur linéaire en Réels ($X_B==0$, $X_C==1$):
$X_D== 1, X_E == 0$
Cout: 2
(tous entiers -> solution (mais pas meilleure) ->
pas mis dans les NA -> fin)

La solution est le nœud vert (première solution rencontrée non remplacée).

$X_B == X_C == 1$

$X_D == X_E == 0$

Coût: 2