# Optimisation des communications avec diffusion et localisation

Vincent Loechner
Laboratoire de Recherche en Informatique de Metz
loechner@lrim.sciences.univ-metz.fr

Catherine Mongenet
Informatique et Calcul Parallèle de Strasbourg
mongenet@icps.u-strasbg.fr

http://icps.u-strasbg.fr/

## Plan de l'exposé

- Définitions
- Communications nécessaires pour une dépendance donnée :
  - localisation
  - diffusion
- Algorithme de recherche de transformations espace-temps efficaces
- Exemple
- Conclusion

# Systèmes d'équations récurrentes affines paramétrés

Le système dépend d'un vecteur de paramètres $p \in \mathbb{Z}^m$.

Il est composé d'un ensemble d'égalités de la forme :

$$X[z] = f(..., Y[g(z,p)], ...) \qquad z \in \mathcal{D}(p)$$

où $g(z,p) = Rz + Qp + r$ est la fonction d'index.

$R$ et $Q$ sont des matrices constantes, $r$ est un vecteur constant.

# Modélisation des dépendances

$$\text{pour } z \in D(p), \qquad X[z] = \ldots\ Y[g(z,p)]\ \ldots \qquad\qquad (Eq)$$

**L'ensemble d'utilisation**

$$\mathrm{Util}_{Eq,Y}(z_0, p) = \{z \in D(p) \mid g(z,p) = z_0\}$$

est l'ensemble des points de calcul utilisant $Y[z_0]$.

⇒ Chaque ensemble d'utilisation est un polyèdre convexe paramétré.

⇒ Ses sommets sont des fonctions affines de $z_0$ et de $p$, définies par domaine de validité sur $z_0$ et $p$.

**L'ensemble d'émission**

$$\begin{aligned}\mathrm{Emit}_{Eq,Y}(p) &= \{z_0 \mid \exists z \in D(p), g(z,p) = z_0\}\\ &= g(D(p), p)\end{aligned}$$

est l'ensemble des points à l'origine des dépendances.

# Modélisation des dépendances

$U$ est la matrice colonne des vecteurs de base des ensembles d'utilisation.

$E$ est la matrice colonne des vecteurs de base de l'ensemble d'émission.

$U$ et $E$ sont calculés à partir de la mise sous forme normale de Hermite de la matrice $R$ :

$$R \cdot [V|U] = [E|0]$$

$V$ est appelée la matrice complémentaire de $U$.

## Transformation espace-temps affine par variable

- **Fonction de cadencement**

$$\boxed{t_X(z) = \lambda_X z + \alpha_X}$$

Contraintes causales :

Lorsqu'on a une dépendance de la forme $X[z] = \ldots\ Y[g(z,p)]\ \ldots$ , $Y$ doit être calculé avant $X$ :

$$t_Y(g(z,p)) \prec t_X(z)$$

- **Fonction d'allocation**

$$\boxed{\mathrm{alloc}_X(z) = \sigma_X z + \gamma_X}$$

**Contrainte de validité :** $\begin{pmatrix} \lambda_X \\ \sigma_X \end{pmatrix}$ doit être *full-row-rank.*

**Objectif :**

Trouver un ensemble de transformations espace-temps valides, et optimales en terme de communication (nombre et type).

# Communications

**Vecteur de communication :**

- un vecteur de communication est l'image d'une dépendance par les fonctions d'allocation
- Considérons la dépendance de $Y$ vers $X$:

$$\text{pour } z \in D(p), \qquad X[z] = \ldots\ Y[g(z,p)]\ \ldots$$

  Le vecteur de communication résultant sur l'espace des processeurs est :

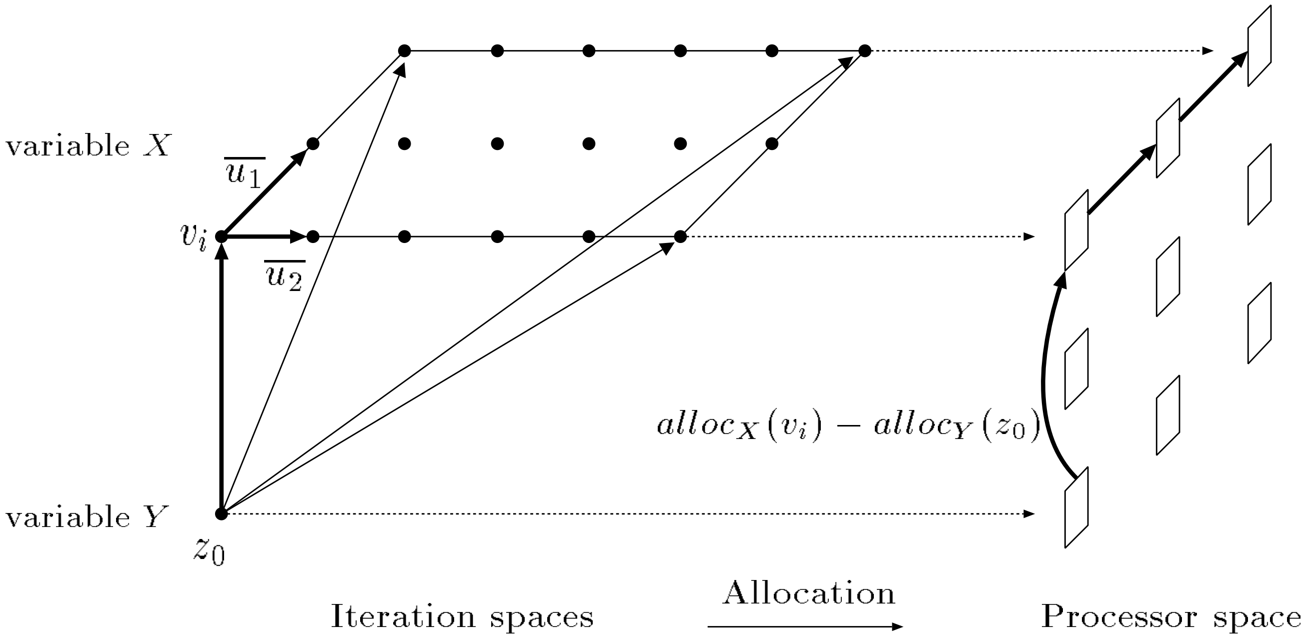
$$\text{alloc}_X(z) - \text{alloc}_Y(g(z,p))$$

  Il n'y a pas de communication si $\text{alloc}_X(z) = \text{alloc}_Y(g(z,p))$ pour tout $z \in D(p)$

$$\Leftrightarrow (\sigma_X - \sigma_Y \cdot R) \cdot \mathcal{B} = 0 \qquad \text{où } \mathcal{B} \text{ est une base de } \mathcal{D}(p)$$

- Problème : si on écrit toutes les contraintes de ce type pour un problème donné, la solution contient souvent un seul processeur.

# Propagation locale




Si $\mathrm{alloc}_X(v_i) - \mathrm{alloc}_Y(z_0)$ est constant pour tout point d'émission $z_0$, la propagation est locale.

# Propagation locale : contraintes sur les transformations

- contrainte sur l'allocation

  $v_i$ est un sommet entier de l'ensemble d'utilisation. C'est une fonction affine de $z_0$ :

  $$v_i = A_i z_0 + h(p)$$

  La contrainte nécessaire et suffisante pour que le premier vecteur de communication soit constant est :

  $$(\sigma_X \cdot A_i - \sigma_Y) \cdot E = 0$$

  où $E$ est la matrice d'émission.

- contrainte sur le cadencement

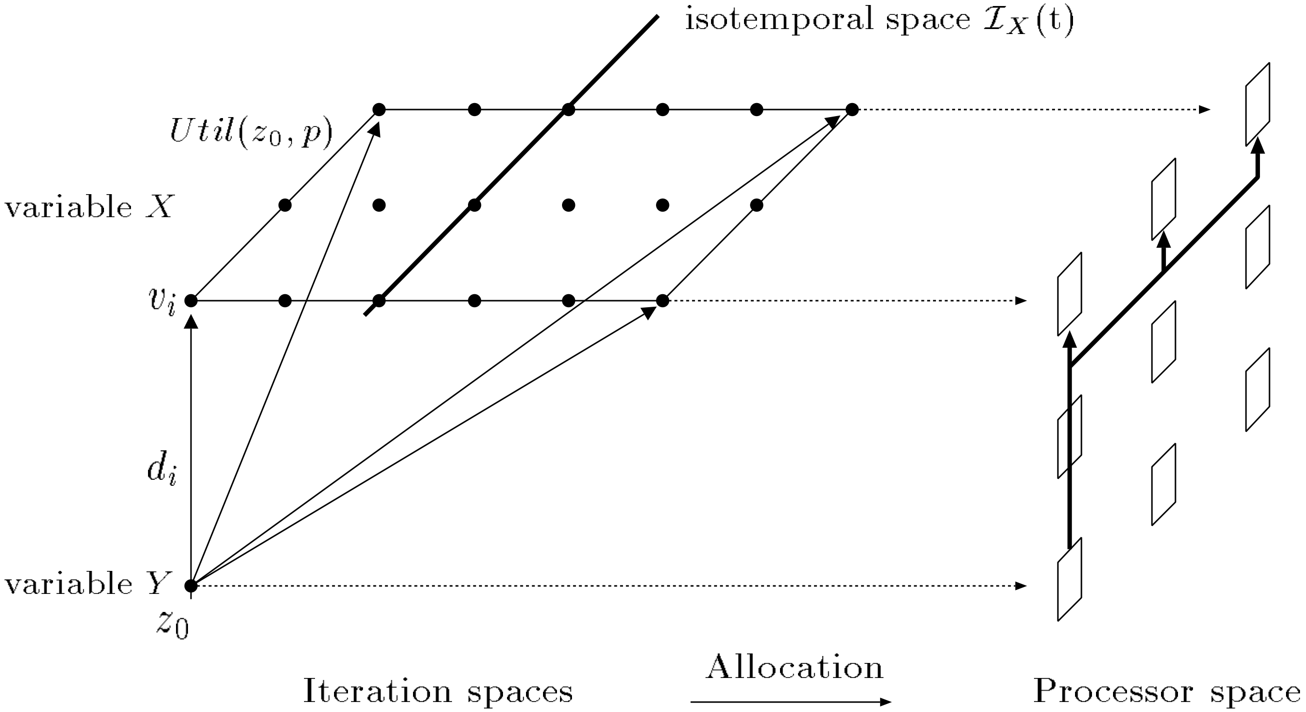
  La contrainte nécessaire et suffisante pour que $v_i$ soit le premier point actif de l'ensemble d'utilisation est :

  $$\lambda_X \, \overline{u_j} \succ 0$$

  pour tout $\overline{u_j}$ vecteur directeur des arêtes de l'ensemble d'utilisation en $v_i$.

## Diffusion

Pour qu'une "vraie" diffusion (non anticipée) soit nécessaire, il faut que la première activation de l'ensemble d'utilisation corresponde à tous les points d'une arête.



# Diffusion : contraintes sur le cadencement

Soient :

$v_i$ un sommet d'un ensemble d'utilisation.

$\overline{U}$ la matrice contenant les vecteurs directeurs des arêtes issues de ce sommet.

Il y a diffusion si :

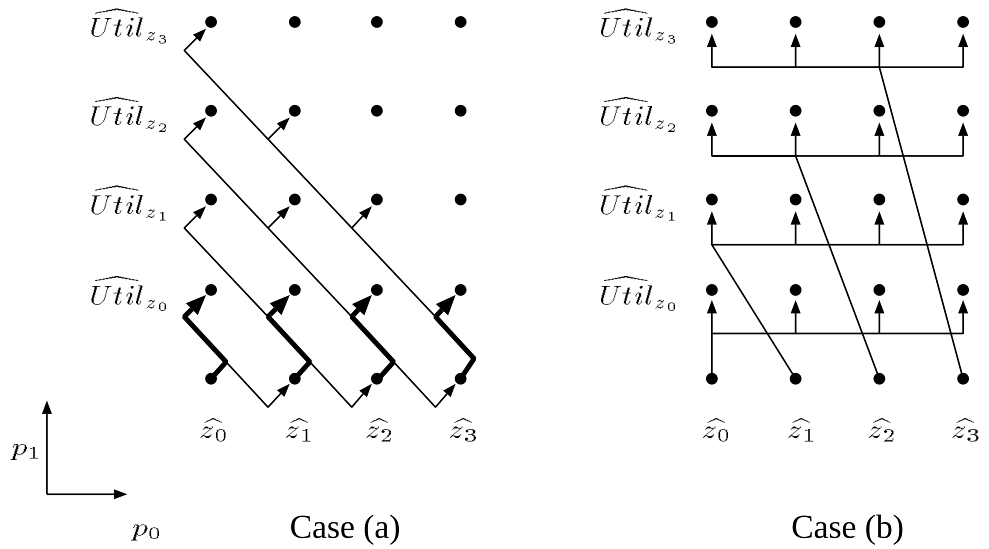
$\overline{U}$ peut être décomposée en deux sous-matrices $\overline{U_1}$ et $\overline{U_2}$ telles que :

$$\begin{cases} \lambda_X \; \overline{u_{1,j}} = 0 & \forall \; \overline{u_{1,j}} \text{ vecteur colonne de } \overline{U_1} \\ \lambda_X \; \overline{u_{2,j}} \succ 0 & \forall \; \overline{u_{2,j}} \text{ vecteur colonne de } \overline{U_2} \end{cases}$$

La dimension de la diffusion est : $Rank(\overline{U_1})$

# Diffusion locale

Un des processeurs récepteurs se trouve toujours à la même distance du processeur émetteur.




Contrainte sur l'allocation : il existe une matrice constante $M$ telle que

$$\sigma_X \cdot (V + U \cdot M) = \sigma_Y \cdot E$$

où $U$ la matrice d'utilisation, $V$ la matrice complémentaire à $U$, et $E$ est la matrice d'émission.

## Diffusion anticipée

Contrainte sur l'allocation :

$$\sigma_X \cdot U \neq 0$$

où $U$ est la matrice d'utilisation.

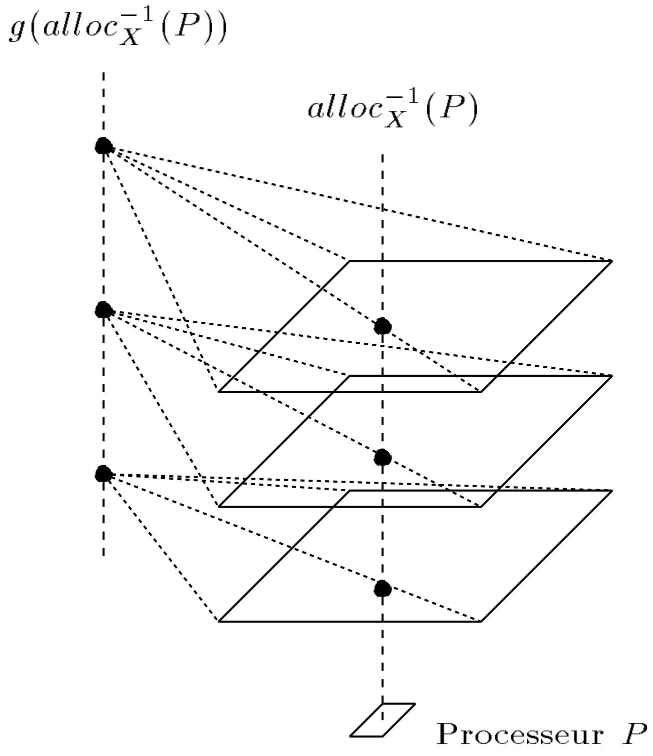
Coût en mémoire : le nombre maximal de données à stocker sur le processeur P correspond au nombre d'ensembles d'utilisation projetés sur $P$.

$\mathrm{alloc}_X^{-1}(P)$ correspond à l'ensemble des points du domaine d'itération projetés sur $P$.

L'image par $g$ de ces points donne les points d'émission "utiles" à $P$, et donc le nombre maximal de données stockées par $P$.

Le coût mémoire maximal pour tout processeur $P$ est donc : $EP(g(\mathrm{alloc}_X^{-1}(P)))$.

# Diffusion anticipée



# Diffusion normale

Diffusion selon les axes du repère des processeurs.

La contrainte sur l'allocation est :

$$l_1 = Rank(\sigma_X \cdot U)$$

avec $l_1$ le nombre de lignes non nulles de la matrice $\sigma_X \cdot U$.

$\sigma_X \cdot U$ est l'image sur l'espace des processeurs des vecteurs d'utilisation.

Le rang de cette matrice est la dimension de l'espace des processeurs d'utilisation, donc la dimension maximale de diffusion.

Les lignes de $\sigma_X$ sont les vecteurs de base de l'espace des processeurs virtuels : ils définissent les axes du repère. Une ligne nulle de la matrice $\sigma_X \cdot U$ signifie qu'il n'y a pas de communication le long de l'axe correspondant.

Si $l_1$ est le nombre de lignes non-nulles de cette matrice alors la diffusion ne peut se faire que sur $l_1$ directions normales.

Pour avoir une diffusion uniquement sur des directions normales, il faut donc que $l_1$ soit égal au rang de $\sigma_X \cdot U$.

## Recherche de transformations efficaces

- classification des dépendances selon la dimension des ensembles d'utilisation et d'émission,
- construction d'un arbre de solutions
  - chaque niveau correspond à une dépendance,

  création des fils d'un noeud :

  - application de contraintes correspondant à un type de communication : diffusion, diffusion anticipée, localisation des ensembles d'utilisation, propagation locale
  - noeud sans contraintes supplémentaires pour s'assurer qu'une solution existe

  transmission de père en fils de :

  - nouvelles contraintes sur le cadencement et l'allocation
  - évaluation des coûts des communications résiduelles

# Exemple

Le système d'équations récurrentes suivant dépend d'un paramètre $N$ :

$$\boxed{\begin{array}{ll} X[i,j,k] = f(Y[j,i-1,0]) & D_{(1)} = \{(i,j,k) \mid 1 \leq i \leq N, 0 \leq j,k \leq N\} \\ Y[i,j,k] = f'(X[0,0,k-1]) & D_{(2)} = \{(i,j,k) \mid 0 \leq i,j \leq N, 1 \leq k \leq N\} \end{array}}$$

Ensembles d'utilisation :

- première dépendance: $X[i,j,k] \leftarrow Y[j,i-1,0]$

$$\mathrm{Util}_{(1)}(i_0,j_0,k_0,N) = \{\, (j_0+1, i_0, k) | \, 0 \leq k \leq N\}$$

- deuxième dépendance: $Y[i,j,k] \leftarrow X[0,0,k-1]$

$$\mathrm{Util}_{(2)}(i_0,j_0,k_0,N) = \{\, (i,j,k_0+1) | \, 0 \leq i \leq N, 0 \leq j \leq N\}$$

# Allocation

$\rightarrow$ espace des processeurs de dimension 2.

$$\begin{aligned}
\mathrm{alloc}_X(\begin{pmatrix} i \\ j \\ k \end{pmatrix}) &= \begin{pmatrix} \sigma_{X1,1} & \sigma_{X1,2} & \sigma_{X1,3} \\ \sigma_{X2,1} & \sigma_{X2,2} & \sigma_{X2,3} \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix} + \begin{pmatrix} \gamma_{X1} \\ \gamma_{X2} \end{pmatrix} \\
&= \begin{pmatrix} \sigma_{X1} & \sigma_{X2} & \sigma_{X3} \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix} + \gamma_X
\end{aligned}$$

# Cadencement

$\rightarrow$ cadencement mono-dimensionnel.

$$t_X(\begin{pmatrix} i \\ j \\ k \end{pmatrix}) \quad = \quad \begin{pmatrix} \lambda_{X1} & \lambda_{X2} & \lambda_{X3} \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix} + \alpha_X$$

Les contraintes causales sont :

$$\begin{cases} \alpha_X = \alpha_Y \\ \lambda_{X1} > 0 \\ \lambda_{Y3} > 0 \\ \lambda_{X1} \geq \lambda_{Y2} \geq 0 \\ \lambda_{X2} \geq \lambda_{Y1} \geq 0 \\ \lambda_{Y3} \geq \lambda_{X3} \geq 0 \end{cases}$$

# Contraintes liées à chaque stratégie

| dep. | strategy | new constraints |
|---|---|---|
| (1) | true broadcast (case 1.) | $\lambda_{X3} = 0$ |
| | local broadcast (case 1b.) | $\exists m_1 \in \mathbb{Z}, \sigma_{X1} + m_1 \sigma_{X3} = \sigma_{Y2}$<br>$\exists m_2 \in \mathbb{Z}, \sigma_{X2} + m_2 \sigma_{X3} = \sigma_{Y1}$ |
| | strong-removal and elimination of all comm. (case 2.) | $\sigma_{X1} = \sigma_{Y2}$<br>$\sigma_{X2} = \sigma_{Y1}$<br>$\sigma_{X3} = 0$<br>$\gamma_Y = \gamma_X + \sigma_{Y2}$ |
| | strong-removal (case 3.) | $\sigma_{X1} = \sigma_{Y2}$<br>$\sigma_{X2} = \sigma_{Y1}$<br>$\sigma_{X3} = 0$ |
| | local propagation ($v_1$) (case 4.) | $\lambda_{X3} > 0$<br>$\sigma_{X1} = \sigma_{Y2}$<br>$\sigma_{X2} = \sigma_{Y1}$ |
| (2) | true broadcast of dimension 2 (case 1.) | $\lambda_{Y1} = 0$<br>$\lambda_{Y2} = 0$ |
| | true broadcast of dimension 1 (case 1.) | $\lambda_{Y1} = 0$<br>or<br>$\lambda_{Y2} = 0$ |
| | local broadcast (case 1b.) | $\exists m'_1, m'_2 \in \mathbb{Z},$<br>$\sigma_{Y3} + m'_1 \sigma_{Y1} + m'_2 \sigma_{Y2} = \sigma_{X3}$ |
| | strong-removal (case 2. or 3.) | $\sigma_{Y1} = 0$    $\nexists\ \sigma_Y$ full row rank.<br>$\sigma_{Y2} = 0$    $\Rightarrow$ The strong condition<br>$\sigma_{Y3} = \sigma_{X3}$    can not be satisfied |
| | local propagation ($v'_1$) (case 4.) | $\lambda_{Y1} > 0$<br>$\lambda_{Y2} > 0$<br>$\sigma_{X3} = \sigma_{Y3}$ |

## Solution 1 : avec diffusion

Toutes les contraintes de diffusion sont compatibles.

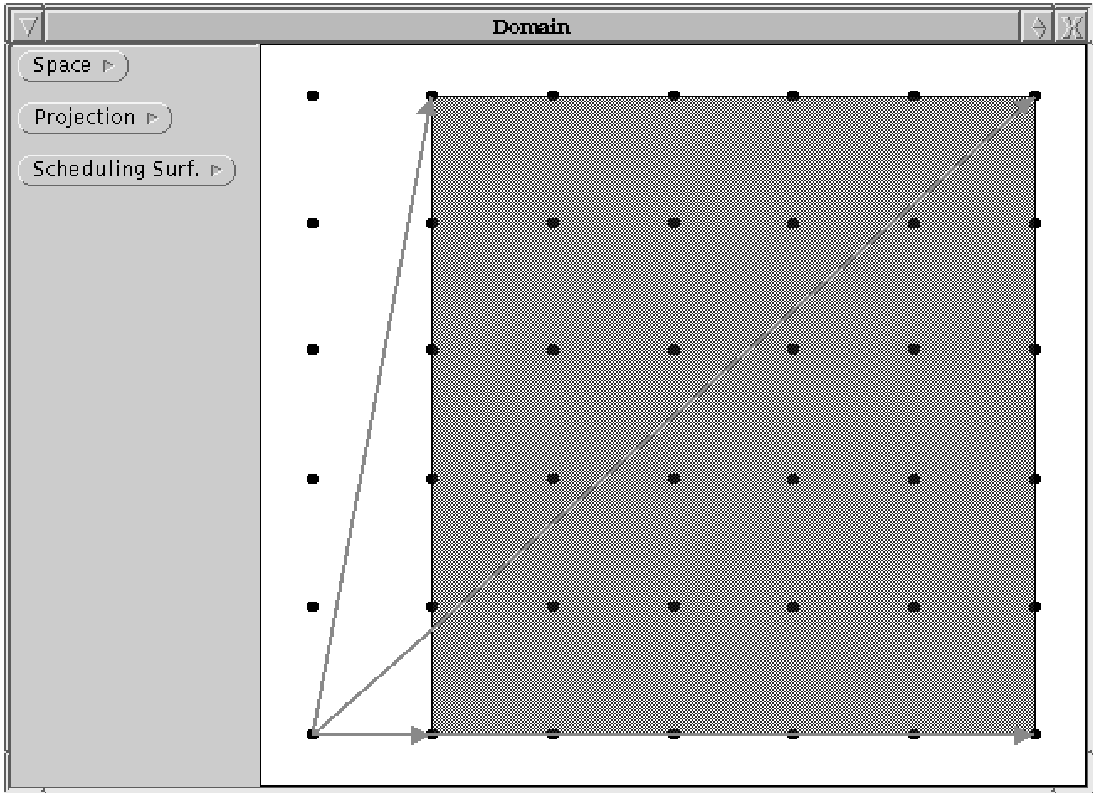
La solution suivante vérifie toutes les contraintes :

$$T_X = \begin{pmatrix} \lambda_X \\ \sigma_X \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \qquad \gamma_X = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

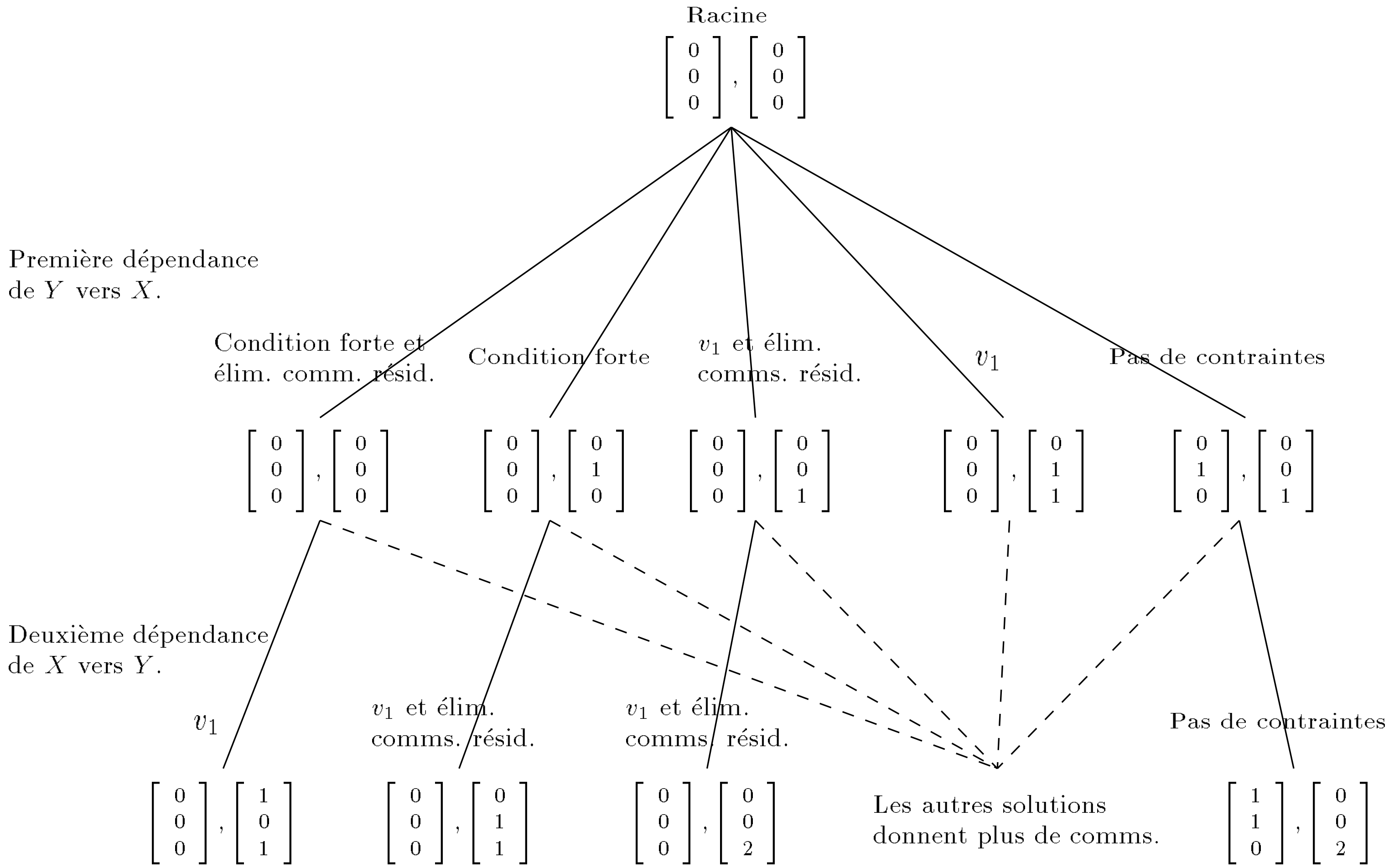
$$T_Y = \begin{pmatrix} \lambda_Y \\ \sigma_Y \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \qquad \gamma_Y = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\alpha_X = \alpha_Y = 0$$

# Solution 1



# Solution 2 : sans diffusion



## Solution 2 : sans diffusion

La meilleure solution est la feuille de gauche de l'arbre.

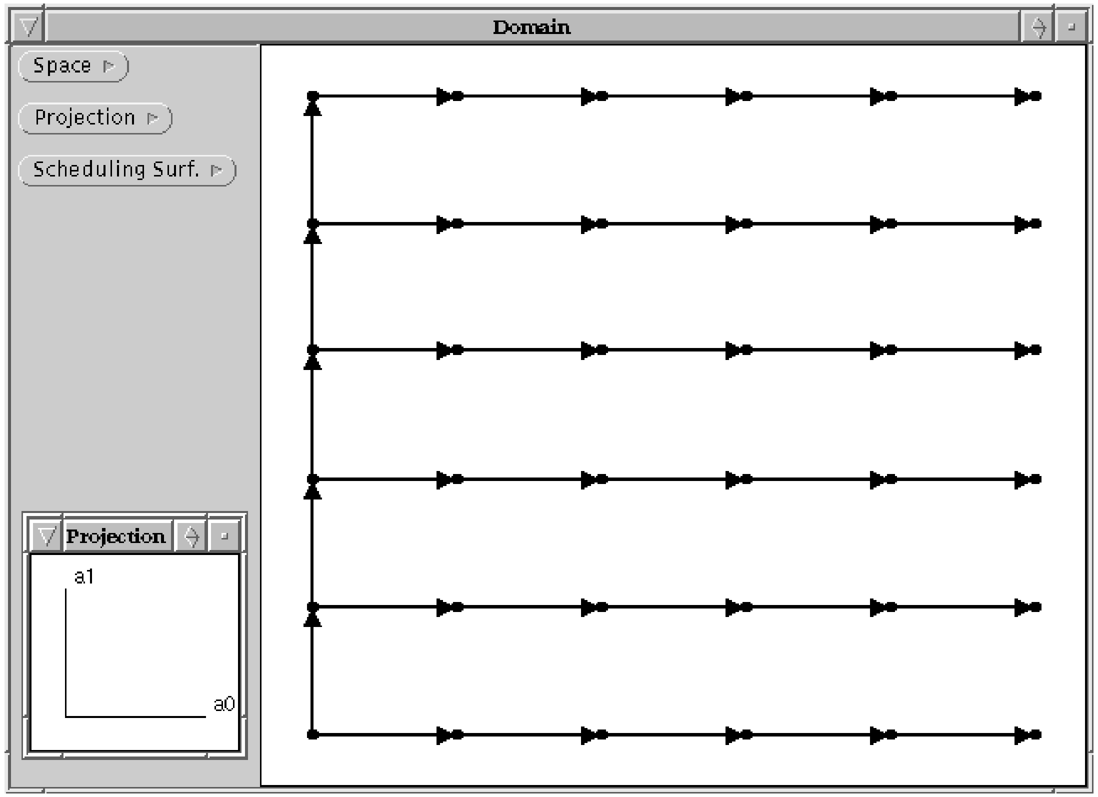
Une solution numérique possible est :

$$T_X = \begin{pmatrix} \lambda_X \\ \sigma_X \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \qquad \gamma_X = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$T_Y = \begin{pmatrix} \lambda_Y \\ \sigma_Y \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \qquad \gamma_Y = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\alpha_X = \alpha_Y = 0$$

# Solution 2



## Perspectives

- l'implémentation est en cours
- applications en programmation de FPGA ou en conception de circuits dédiés
- raffiner l'algorithme de recherche de transformations espace-temps
  - optimisation de la construction de l'arbre
  - évaluation du coût des communications plus fine
- extensions de PolyLib V4
  - recherche de sommets *entiers* de polytopes paramétrés
  - polyèdres paramétrés non linéaires : $A(p)\ x \geq Bp + b$